

**UNIVERSIDAD CENTRAL “MARTA ABREU” DE LAS VILLAS
FACULTAD DE MATEMÁTICA, FÍSICA Y COMPUTACIÓN
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**



**AMBIENTE DE ENSEÑANZA-APRENDIZAJE INTELIGENTE PARA LA
PROGRAMACIÓN LÓGICA**

Tesis presentada en opción al grado científico de Doctor en Ciencias Técnicas

Autor: MSc. Lydia Rosa Ríos Rodríguez

Tutor: Dr. Mateo Lezcano Brito

Santa Clara

2009

**UNIVERSIDAD CENTRAL “MARTA ABREU” DE LAS VILLAS
FACULTAD DE MATEMÁTICA, FÍSICA Y COMPUTACIÓN
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**



**AMBIENTE DE ENSEÑANZA-APRENDIZAJE INTELIGENTE PARA LA
PROGRAMACIÓN LÓGICA**

Tesis presentada en opción al grado científico de Doctor en Ciencias Técnicas

Autor: MSc. Lydia Rosa Ríos Rodríguez

Santa Clara

2009

Agradecimientos

Quiero agradecer de manera especial a:

Mis padres y mis hermanos porque siempre puedo contar con ellos.

A Erasmo por su ayuda incondicional, comprensión y amor durante todos estos años.

A mi tutor, el Dr. Mateo Lezcano Brito por su tiempo, su paciencia y su dedicación.

A Mercedes, por el cariño y todo el apoyo brindado.

A todos mis compañeros de departamento por su solidaridad permanente.

A Bécquer y a Nelsy, que jamás han tenido un NO para mí.

A los profesores de inglés Magalys y Evelio por sus amables colaboraciones.

A mis amigos Yunia, Tania, Catalina, Maira, Néstor, Adilén, Osvaldito y Carlos porque a pesar de sus complicaciones siempre estuvieron presentes cuando los necesité.

Al Dr. Pedro Fuentes por la confianza que siempre ha tenido en mí.

A Mirelis, Elsy, Yandira, Aldo, Maité, Aljadis, Addislexy, Fardales, Roxana, Aurelio y Liset por sus importantes contribuciones.

A todos los estudiantes de Ingeniería Informática que cursaron el cuarto año durante el curso 2008-2009.

A Lutgardo, por sus decisivos aportes.

Al Dr. Armando Boullosa, Héctor y Bernardo por estar siempre dispuestos a ayudarme.

A los Doctores Tomás Crespo, María M. García, Zenaida García, Carlos Morell, Ricardo Grau, María del Carmen, Natalia y AbelRodríguez por sus preciadas sugerencias.

A Bismaida, Obregón y Mary, por brindarme sus experiencias de forma desinteresada.

A todos los trabajadores de la Facultad de Ingeniería del Centro Universitario

“José Martí” de Sancti Spiritus.

A Cheché y a Yoly.

Dedicatoria

A mis abuelos Adela y Antonio.

A Dani.

SÍNTESIS

En esta investigación se demuestra la posibilidad de integrar los mapas conceptuales, los agentes inteligentes y las TIC para auxiliar el proceso de enseñanza-aprendizaje de la Programación Lógica. Se propone el Ambiente de Enseñanza-Aprendizaje asistido por computadora para la Programación Lógica (APA-Prolog), con una interfaz basada en mapas conceptuales que incluye como recursos: textos explicativos, simulaciones, entrenadores, tutoriales, evaluadores, un directorio temático y un espacio para el intercambio. Posibilita una navegación libre o asistida. La asistida permite tomar en cuenta los conocimientos previos de los estudiantes. Se presenta una metodología para la creación de ambientes de enseñanza-aprendizaje asistidos por computadora.

El ambiente y los recursos propuestos han sido sometidos al criterio de expertos y su aplicación en el proceso de enseñanza-aprendizaje de la asignatura Programación Descriptiva, al juicio de los estudiantes de la carrera Ingeniería Informática en el Centro Universitario de Sancti Spiritus “José Martí Pérez”.

SYNTHESIS

This research gives the possibility of integrating conceptual maps, the intelligent agents and the TIC, what contributes to help the teaching-learning process of Logic Programming. It is proposed to the teaching-learning environment assisted by computer for Logic Programming (TLP-Prolog), with an interface based on conceptual maps that include as resources: applicative texts, simulations, trainers, tutorials, evaluators, a theme directory and a space for interchange. It scopes a free and assisted surfing. The former (assisted) permits to take into consideration the students' previous knowledge in the main concepts represented in the environment. It proposes a methodology for the creation of a teaching-learning assisted system supported by computer, based on conceptual maps and with the possibility of taking into account the students' knowledge.

The environment and the resources that it proposes have been studied using qualitative methods, based on experts' criteria, as well as quantitative ones to measure students' criteria after putting it into practice in the teaching learning process of the subject Descriptive Programming in the speciality of Informatic Engineering in "José Martí Pérez" University Centre in Sancti Spiritus.

ÍNDICE

	Pág.
<u>INTRODUCCIÓN.....</u>	<u>1</u>
1. LOS SISTEMAS MULTIAGENTES Y LA PROGRAMACIÓN LÓGICA. VISION PARA LA ENSEÑANZA USANDO LA WEB.....	10
<u>1.1 Agentes de software.....</u>	<u>11</u>
<u>1.1.1 Agentes Inteligentes (AI).....</u>	<u>12</u>
<u>1.1.2 Herramientas para la construcción de Agentes Virtuales Inteligentes (IVA).....</u>	<u>14</u>
<u>1.1.3 Los Agentes Inteligentes en ambientes de enseñanza –aprendizaje.....</u>	<u>15</u>
<u>1.2 Sistemas Multiagentes (SMA).....</u>	<u>21</u>
<u>1.2.1 Descripción de los sistemas multiagentes.....</u>	<u>23</u>
<u>1.2.2 Sistemas Multiagentes Inteligentes.....</u>	<u>24</u>
<u>1.2.3 Sistemas Multiagentes en la Enseñanza.....</u>	<u>26</u>
<u>1.3 Los mapas conceptuales.....</u>	<u>28</u>
<u>1.3.1 Los mapas conceptuales en la enseñanza.....</u>	<u>29</u>
<u>1.3.2 Los mapas conceptuales y las tecnologías de la información.....</u>	<u>31</u>
<u>1.4 La enseñanza de la Programación Lógica.....</u>	<u>32</u>
<u>1.4.1 La enseñanza de la Programación Lógica en Cuba.....</u>	<u>34</u>
<u>1.5 Conclusiones Parciales.....</u>	<u>40</u>
2. APA-Prolog: un ambiente de enseñanza-aprendizaje asistido por computadora para la Programación Lógica.....	42
<u>2.1 Características del Ambiente de Enseñanza-Aprendizaje para la Programación Lógica (APA-Prolog).....</u>	<u>42</u>
<u>2.2 Arquitectura básica de un agente.....</u>	<u>45</u>
<u>2.2.1 Agente Adis.....</u>	<u>45</u>
<u>2.2.2 Agente Tivo.....</u>	<u>49</u>
<u>2.2.3 Agente Teo.....</u>	<u>50</u>
<u>2.2.4 Agente Tica.....</u>	<u>57</u>
<u>2.2.5 Agente Eva.....</u>	<u>60</u>
<u>2.2.5.1 Evaluadores.....</u>	<u>61</u>
<u>2.3 Espacio para el intercambio.....</u>	<u>62</u>
<u>2.4 Directorio Temático.....</u>	<u>64</u>
<u>2.5 Modelo para la solución cooperativa.....</u>	<u>67</u>
<u>2.6 Conclusiones Parciales.....</u>	<u>71</u>
3. METODOLOGIA PARA CONSTRUIR SISTEMAS MULTIAGENTES INTELIGENTES BASADOS EN WEB. ALGUNAS APLICACIONES.....	73

<u>3.1 Experiencia del empleo de APA-Prolog en el CUSS.....</u>	<u>73</u>
<u>3.2 Metodología para la creación de ambientes de enseñanza-aprendizaje usando mapas</u>	
<u>conceptuales que toman en cuenta los conocimientos previos del estudiante.....</u>	<u>78</u>
<u>3.2.1 Generalidades.....</u>	<u>78</u>
<u>3.2.2 Metodología.....</u>	<u>79</u>
<u>3.2.3 Empleo de la metodología en otras áreas de conocimiento.....</u>	<u>83</u>
<u>3.3 Validación estadística de la efectividad de APA-Prolog.....</u>	<u>84</u>
<u>3.4 Análisis de los resultados valorados por expertos.....</u>	<u>86</u>
<u>3.4.1 Procesamientos estadísticos utilizados.....</u>	<u>87</u>
<u>3.4.2 Valoraciones de los expertos.....</u>	<u>87</u>
<u>3.5 Conclusiones Parciales.....</u>	<u>91</u>
<u>CONCLUSIONES Y RECOMENDACIONES.....</u>	<u>92</u>
<u>REFERENCIAS BIBLIOGRÁFICAS.....</u>	<u>93</u>
<u>PRODUCCIÓN CIENTÍFICA DEL AUTOR SOBRE EL TEMA DE LA TESIS.....</u>	<u>100</u>
<u>ANEXOS.....</u>	<u>103</u>

INTRODUCCIÓN

La actuales generaciones transitan una época matizada por la información donde se habla comúnmente, en escenarios educativos diversos, del desarrollo de las potencialidades del ser humano, de la autonomía del alumno y de la necesidad de aprender a aprender (López, 2007). El fácil acceso a la información y su distribución por medios electrónicos multiplica el impacto formativo de las instituciones de educación. La mayor interacción entre las comunidades académicas permite un proceso continuo de mejoramiento de la calidad educativa; la apertura a la interacción mundial potencia los procesos de transformación de las instituciones educativas (Almeida, 2007). Es por ello que hacer de la virtualización un instrumento al servicio de la Nueva Universidad es uno de los propósitos de la dirección de Informatización del Ministerio de Educación Superior en Cuba.

En este contexto no debe perderse de vista, que diseñar recursos para la enseñanza bajo un esquema de “talla única” conllevaría a proveer a los estudiantes de recursos didácticos no necesariamente acordes a sus características, conocimientos previos o nivel de desarrollo cognitivo, en otras palabras a sus necesidades individuales de formación (Berlanga y García, 2004). Otro elemento a tomar en cuenta es la forma en que se organiza el contenido. Muchas veces los estudiantes o cualquier persona que desea investigar se sienten defraudados cuando navegan por la Web y se declaran perdidos dentro de la tela de araña en que se transforma la información cuando redundan los hipervínculos y no existe una manera clara de navegar.

Los Sistemas Hipermedia Adaptativos (SHA) y las Web adaptativas representan un área de interés creciente de investigación en las Tecnologías de la Información y las Comunicaciones (TIC). Los investigadores de los SHA se enfocan al estudio de teorías, técnicas e innovaciones tecnológicas que permitan la personalización y adaptación de la información a las necesidades específicas de cada usuario (Grimón, 2008). Diversas técnicas de Inteligencia Artificial (IA), entre las que descuellan los agentes inteligentes, se emplean con estos fines.

Los mapas conceptuales (Novak y Gowin, 1988), que fueron inicialmente concebidos sólo para organizar el conocimiento, también constituyen hoy en día un medio de navegación

que, además ayuda al aprendizaje visual. Las imágenes creadas con esta técnica no sólo transmiten información básica sino que permiten ver las relaciones, estructuras, modelos y características del dominio en cuestión.

Las potencialidades que hoy ofrecen las computadoras son indiscutibles como también lo es la necesidad de programarlas para lograr que hagan lo que realmente se necesita de ellas. Desde los albores de la computación y a partir del mismo momento en que surgieron los primeros lenguajes de programación, programar siempre se ha considerado una tarea difícil. Esta afirmación es realidad aún en el momento actual a pesar de los avances logrados en las técnicas y mecanismos de puesta a punto de las que hacen gala los ambientes actuales de desarrollo integrados.

Dado el hecho anterior, la enseñanza de la programación resulta una tarea difícil ya que no sólo se trata de enseñar instrucciones de un determinado lenguaje que permiten codificar los algoritmos, sino que debe enseñarse a pensar en forma algorítmica para de esa manera poder resolver el problema real que se desea modelar.

Por estas razones, los profesores que tienen la responsabilidad de enseñar a programar pueden estar ante dos situaciones:

- La primera y más fácil es enseñar un lenguaje que pertenece a un paradigma de programación conocido. En ese caso el nuevo lenguaje sigue estrategias comunes dictadas por ese paradigma y las diferencias no serán marcadamente notables,
- La segunda situación se tiene cuando se enseña un nuevo paradigma. En este caso no sólo se está enseñando un nuevo lenguaje sino, además, una forma distinta de pensar acerca de la solución del problema, lo que trae por consecuencia que el pensamiento algorítmico del paradigma previo y el nuevo no sean compatibles, a tal extremo, que pueda interferir en la nueva forma de pensar. Según (Lezcano, 1998) esta es la situación a la que se enfrentan docentes y estudiantes de las carreras Ciencias de la Computación e Ingeniería Informática de las universidades cubanas cuando comienzan a estudiar el paradigma descriptivo a partir del cuarto año de sus respectivas carreras.

En esta investigación se constató que muchas universidades que desarrollan la carrera Ciencia de la Computación (o carreras afines) también contemplan la Programación Lógica

en sus planes de estudio; algunas en el pregrado, otras en el postgrado, en algunos casos como asignaturas obligatorias y en otros como opcionales. Entre estos centros se pueden citar los siguientes: la Universidad de Washington¹ (EE.UU), la Universidad de Sevilla² (España), la Universidad Carlos III de Madrid³ (España), la Universidad de Stanford⁴ (EE.UU), la Universidad Nacional Autónoma de México⁵, la Universidad Nacional de La Plata⁶ (Argentina), la Universidad de Concepción⁷ (Chile) y la Universidad Nacional de Villa María⁸ (Argentina).

En Cuba, la Programación Lógica forma parte del sistema de contenidos de los planes de estudios C y D de las carreras Ingeniería Informática y Ciencias de la Computación. Por ello, los estudiantes deben conocer los fundamentos y características de este paradigma de programación, sus posibilidades y limitaciones.

En los últimos años, se registra una gran cantidad de aplicaciones basadas en el paradigma de la Programación Lógica, entre ellas (Vasconcelos y Robertson, 2003), (Wilson, 2005), (Demoen, 2005), (Christiansen, 2005), (Rocha, 2005) (Schindlauer, 2005) y (Casali, 2008). Por la facilidad que encierra, como forma de representar el conocimiento, se emplea con frecuencia para la implementación de sistemas multiagentes y agentes inteligentes (Torroni, 2003), (Fernández, 2005), (Das, 2008), (Díaz, 2009) y (Hjelmbom, 2009). Todo esto genera una reflexión importante: si no se trabaja por atenuar las dificultades detectadas en la enseñanza de la Programación Lógica, los egresados no serán capaces de desempeñarse como se espera de ellos.

La autora de esta tesis ha constatado, a partir de entrevistas e intercambio con profesores, que en el afán de mejorar la enseñanza se recurre a diferentes técnicas que, algunas veces, resultan un tanto empíricas. Una mejor solución debe partir de un estudio que establezca las insuficiencias que se deben resolver para después definir las acciones y crear las

¹ www.cs.washington.edu

² <http://www.cs.us.es>

³ <http://www.uc3m.es>

⁴ <http://www.stanford.edu>

⁵ www.unam.mx

⁶ <http://www.unlp.edu.ar>

⁷ <http://www.inf.udec.cl>

⁸ <http://webs.satlink.com/usuarios/l/leibnitz/Licenc.htm>

herramientas que auxiliarán en la tarea de resolver las dificultades encontradas. Con este propósito, esta investigación parte de un estudio realizado por el grupo de Informática Educativa de la Universidad Central de Las Villas (UCLV) (Lezcano, 1998).

El estudio mencionado permitió identificar las principales dificultades que presentan los estudiantes en el proceso de enseñanza-aprendizaje de la Programación Lógica, entre las que cabe destacar las siguientes (se detallarán en el capítulo 1):

1. No saben describir el problema planteado.
2. Escriben reglas con antecedentes no definidos.
3. No expresan claramente lo que se necesita a través de una relación.
4. Hacen mal uso de los términos compuestos o estructuras.
5. No realizan un manejo adecuado de las listas.
6. Incapacidad para abstraer la esencia de un problema.

Mediante esta investigación se pudo constatar la escasez de recursos informáticos que auxilien el proceso de enseñanza de la Programación Lógica, las encontradas se reducen a sitios Web y tutoriales que, a pesar de su utilidad, reducen su alcance a presentar información mayoritariamente estática. Sólo se encontró una herramienta interactiva, denominada Teaching (Lezcano, 1998) que es uno de los resultados que sirve de antecedente a este estudio.

Teaching tiene la particularidad de estar programado en el mismo paradigma objeto de estudio, la Programación Lógica. Incluye un tutorial estructurado como hipertexto que explica teóricamente los conceptos fundamentales del lenguaje y una herramienta de simulación que dispone de un conjunto de programas ejemplos que facilitan el aprendizaje al mostrar los caminos de ejecución tomados, lo que permite a los estudiantes comprender la forma en que el lenguaje resuelve algunos problemas apreciando los cambios sobre las estructuras de datos utilizadas.

La herramienta Teaching es un excelente medio de enseñanza, sin embargo, posee las limitaciones propias de la época en que se creó (1998): Programado sobre MS-DOS permite solo el trabajo individual, no toma en cuenta las posibilidades de las redes y no

incluye elementos de adaptación. Además, su alcance se reduce a un pequeño conjunto de problemas y conceptos.

Esta situación nos lleva a plantear el siguiente **problema de investigación**:

¿Qué recursos informáticos tienen que ser desarrollados e integrados de manera que la asimilación de los contenidos de la Programación Lógica se ajuste mejor a las características individuales de los estudiantes y en qué medida ello repercute en la calidad del proceso?

Para enfrentar el problema mencionado, se enuncia la siguiente **hipótesis**:

Si se desarrolla un conjunto de recursos informáticos que permita adaptarse al nivel de conocimientos de los estudiantes, se alcanza una enseñanza personalizada que incide positivamente en la calidad del proceso de asimilación de los contenidos de la Programación Lógica.

El **Objetivo General** de esta investigación es:

Concebir y desarrollar un ambiente de enseñanza-aprendizaje asistido por computadora que contenga e integre los recursos computacionales necesarios para incidir positivamente en la calidad del proceso de asimilación de los contenidos de la Programación Lógica en carreras con perfil informático.

Los **Objetivos Específicos** son:

1. Diseñar e implementar una interfaz en forma de mapa conceptual, que muestre y relacione los conceptos fundamentales de la Programación Lógica.
2. Diseñar y programar un conjunto de herramientas que sirvan de auxiliar en la enseñanza de la Programación Lógica. Insertar las herramientas confeccionadas en el mapa conceptual.
3. Diseñar y programar una arquitectura multiagente que proporcione adaptabilidad al Ambiente para la Enseñanza de la Programación Lógica.
4. Evaluar la efectividad y la calidad del ambiente creado.

Entre los Métodos de trabajo científico utilizados se destacan los siguientes:

- Analítico-sintético al descomponer el problema de investigación en elementos por separado y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución de las propuestas.
- Histórico-lógico para el estudio crítico de los trabajos anteriores, de su evolución y para utilizar estos como punto de referencia y comparación de los resultados alcanzados.
- Hipotético-deductivo para la elaboración de la hipótesis de la investigación y para proponer nuevas líneas de trabajo a partir de los resultados parciales obtenidos.
- Modelación para la concepción y el desarrollo de los algoritmos.
- Sistémico para el desarrollo de los diferentes sistemas computacionales y para lograr que los elementos que formen parte de la aplicación real sean un todo que funcione de manera armónica.
- Experimental para comprobar la utilidad de los resultados obtenidos a partir de las propuestas definidas.
- Matemáticos-estadísticos para la validación de los aportes fundamentales de la investigación.
- Coloquial para la presentación y discusión de los resultados en sesiones científicas.

Novedad Científica

El primer aporte que se defiende en este trabajo es la organización del sistema de conocimiento de la asignatura Programación Lógica en forma de mapas conceptuales. Esta organización permite establecer relaciones entre los conceptos y métodos de la Programación Lógica, lo que constituye una vía para mejorar el proceso de enseñanza aprendizaje de esta disciplina.

El segundo aporte lo constituye la arquitectura del sistema multiagente inteligente que toma en cuenta las individualidades cognitivas de los educandos, a través del estado de su conocimiento, para adaptarse a ellas.

El **valor práctico** de este trabajo está dado por los siguientes resultados concretos:

- La obtención de una herramienta, disponible a través de la Intranet del MES o Internet, que posibilita la enseñanza personalizada, ya sea presencial o a distancia, de un tema complejo que se trata en todas las carreras de perfil informático del país. El sistema incluye un conjunto variado de recursos informáticos.

Además, en el desarrollo de esta investigación se han obtenido otras aplicaciones informáticas de alto valor práctico:

- Los simuladores de backtracking, matching y del trabajo con listas, que permiten un aprendizaje activo y por descubrimiento.
- La herramienta para la administración de entrenadores y evaluadores HERAD que puede ser usada en otras asignaturas.
- El directorio temático que facilita el manejo de la bibliografía disponible sobre la programación lógica dentro o fuera del ambiente.

El cuerpo de la **Tesis** se **estructura** en tres capítulos, cuyos contenidos se describen, brevemente, a continuación.

Capítulo 1: "Los sistemas multiagentes y la Programación Lógica. Visión para la enseñanza usando la Web".

En este capítulo se describe qué son los agentes de software, los sistemas multiagentes y cuáles son algunas de sus aplicaciones en la instrucción. Se pasa revista a los mapas conceptuales y sus aplicaciones en la enseñanza. Se hace un estudio de los sistemas adaptativos y de los elementos a tener en cuenta para lograr adaptación. Se hace un análisis de la enseñanza de la Programación Lógica en la carrera Ingeniería Informática, acorde a los planes de estudios vigente en las universidades cubanas (López, 2007).

Capítulo 2: "APA-Prolog: un ambiente de enseñanza-aprendizaje asistido por computadora para la Programación Lógica".

En este capítulo se describen las características generales del Ambiente Para la enseñanza-Aprendizaje de la Programación Lógica (APA-Prolog). Se detallan aspectos de su concepción y diseño. Se describe la arquitectura del sistema multiagente, así como cada uno de los recursos contenidos en el ambiente.

Capitulo 3: "Metodología para construir sistemas multiagentes inteligentes basados en Web. Algunas aplicaciones".

En este capítulo se describe la experiencia del empleo de APA-Prolog en el Centro Universitario de Sancti Spiritus (CUSS). Se presenta una metodología para confeccionar ambientes de enseñanza-aprendizaje usando mapas conceptuales que permitan tomar en cuenta los conocimientos previos del estudiante y se ejemplifica su uso en otras áreas del saber. Se hace un análisis estadístico para validar la efectividad de la herramienta desarrollada y finalmente se describen las valoraciones de los expertos sobre la calidad del producto que se obtuvo.

Capítulo 1

Los sistemas multiagentes y la
Programación Lógica. Visión para la
enseñanza usando la Web

1. LOS SISTEMAS MULTIAGENTES Y LA PROGRAMACIÓN LÓGICA. VISION PARA LA ENSEÑANZA USANDO LA WEB

“Dentro de pocos años, millones de escolares tendrán acceso a algo de lo que disfrutaba el hijo de Filipo de Macedonia, Alejandro, como una prerrogativa real: los servicios personales de un tutor tan bien informado e idóneo como Aristóteles” (Almeida, Febles y Bolaños, 1997). Así comenzaba prediciendo Patrick Suppes, filósofo y matemático de la Universidad de Stanford, en un artículo que apareció en 1966, en la popular revista Scientific American, donde resumía las expectativas y las ideas de ese momento y sostenía que la verdadera función revolucionaria de las computadoras en la educación, se debía a la nueva área de la instrucción asistida por computadora.

Cuarenta años después es común oír hablar de entornos virtuales de aprendizaje, de ambientes virtuales de enseñanza, de ambientes de enseñanza-aprendizaje asistidos por computadora (AEAC) (Ríos, Lezcano, y López, 2008) y de otros términos similares, los cuales son el resultado de la evolución de la enseñanza asistida por computadora, de la introducción de las tecnologías de la información y las comunicaciones en la educación y del empleo de la inteligencia artificial.

Uno de los mayores retos que enfrentan los diseñadores de estos sistemas es brindar una enseñanza apropiada para cada estudiante. Los avances más recientes en este campo sugieren el uso de técnicas como Sistemas Multiagentes para lograrlo.

Por otra parte, los mapas conceptuales son ampliamente utilizados en la enseñanza debido a que permiten dirigir la atención, tanto del estudiante como del profesor, hacia una cantidad reducida de ideas importantes en las que deben concentrarse en cualquier tarea específica del aprendizaje.

En este capítulo se recogen los aspectos más relevantes sobre la tecnología de agentes y los mapas conceptuales aplicados a la educación, así como algunas particularidades de la enseñanza de la Programación Lógica.

1.1 Agentes de software.

La palabra “agente” se escucha con frecuencia en las comunidades actuales dedicadas a las ciencias de la computación y al desarrollo de aplicaciones. En este caso se refiere a agentes de software, los que simulan a los agentes humanos y son el resultado de investigaciones en el campo de la ingeniería de software y la inteligencia artificial.

Del mismo modo que un agente humano tiene ojos, oídos y otros órganos para percibir el medio y manos, boca, piernas y otras partes del cuerpo para actuar sobre él, un agente de software percibe el medio a través de sensores y actúa sobre ese medio a través de acciones (Russell y Norvig, 1995) con la diferencia de que en este caso la percepción y la acción se realizan a través de segmentos de código. En otras palabras, son entidades computacionales que existen físicamente en forma de programas que se ejecutan sobre computadoras (Weiss, 1999).

Técnicamente, los agentes de software exhiben muchos beneficios, entre ellos:

- Modularidad. (Permitiendo adicionar y eliminar elementos sin afectar otros)
- Velocidad. (La ejecución concurrente de programas incrementa la velocidad de ejecución de todo el sistema)
- Otras, tales como mantenibilidad, reusabilidad e independencia de la plataforma (Peña, 2004).

Los agentes no siempre actúan de manera solitaria, también se les puede encontrar formando parte de una sociedad. Por ello (Martin, Gregory y Bradley, 2005) sostienen que poseen una identidad que es muy importante conservar y que está dada no por la forma de su cuerpo sino por la forma en que son vistos por los demás agentes y los usuarios pues la cuestión que define la identidad no es como soy yo sino cómo yo soy visto por los demás.

Al tener vida social, los agentes interactúan entre sí; esto quiere decir que pueden afectar o ser afectados por otros mientras ejecutan sus tareas, por ejemplo, pueden suministrar información que interesa o tal vez confunde a otros agentes. La coordinación es una forma de interacción particularmente importante para el logro de las metas de los agentes y la terminación de sus tareas. Para coordinar sus metas y tareas los agentes tienen que tomar en consideración explícitamente las dependencias entre sus actividades. Las dos formas de

coordinación son: cooperación y competición (Weiss, 1999). En el caso de la cooperación, varios agentes trabajan unidos para alcanzar un objetivo común; es decir, trabajan como un equipo (Grimaldo, Lozano, Barber y Orduna, 2005) cuyo resultado puede fallar o tener éxito. En el caso de la competición, varios agentes trabajan contra otros porque sus objetivos están en conflicto, tratan de maximizar sus propios beneficios y el éxito de uno implica el fallo de otros.

A los agentes se les atribuyen características como autonomía e inteligencia. Decir que son autónomos significa que ellos tienen control sobre su conducta y activación sin la intervención de humanos u otros sistemas (Weiss, 1999). Que sean inteligentes no significa que ellos nunca fallen sino que operan flexible y racionalmente en variadas circunstancias de su medio, en dependencia de sus capacidades de percepción y acción.

1.1.1 Agentes Inteligentes (AI)

Según (Gillies, Robertson y Ballin, 2005) la conducta proactiva y la autonomía en las respuestas al medio distinguen a los agentes inteligentes de los no inteligentes.

En (Hernansáez, 2005) se consideran tres atributos ideales, o sea, que no pueden faltar en un AI, aunque posean otros adicionales. Estos son: autonomía, cooperación y capacidad de aprendizaje. Por autonomía este autor entiende la capacidad de actuar basándose en su experiencia, adaptación a los cambios del entorno, aunque estos sean bruscos, y no necesitar la intervención del usuario humano constantemente. Por cooperación se entiende la interacción y comunicación que tiene el AI con otros agentes para lograr sus metas. Por capacidad de aprendizaje se asume el criterio de que los agentes tienen que aprender a responder bien ante los cambios que ocurran en el ambiente (Shalfield, 2005).

Como características adicionales a los tres atributos ideales, este mismo autor, agrega: conocimiento, persistencia y adaptatividad. Desde la perspectiva que los agentes tienen que ser conscientes de los cambios ocurridos en su ambiente y reaccionar ante esos cambios, funcionar por períodos de tiempo indefinidos y adaptar su comportamiento con el tiempo.

Por su parte (Weiss, 2003) añade a las características anteriores, proactividad, reactividad y movilidad considerando para ello la forma de responder a los cambios y la capacidad de moverse de una computadora a otra o por toda la red.

A partir de los criterios anteriores, puede considerarse un AI como una entidad que se encuentra en un ambiente e interactúa con él, es capaz de tener autonomía y control de sí mismo, puede actuar ante cambios en este ambiente sin perder el objetivo para el cual fue creado, es capaz de interactuar con otros agentes, de hacer lo correcto y de aprender de su experiencia, teniendo en cuenta que a medida que este aprendizaje sea mayor, mayor será su grado de autonomía porque podrá actuar basado en su experiencia ante un suceso en el presente. Puede también, si su función lo requiere, viajar por toda la red y explorarla.

Para diseñar un buen agente hay que pensar en su arquitectura. La arquitectura de un agente define los mecanismos que permiten interconectar los componentes tanto de software como de hardware, que le facilitan al agente comportarse como tal.

Cada arquitectura particular para construir agentes, especifica cómo se descompone en un conjunto de módulos que interactúan entre sí para lograr la funcionalidad requerida. Un aspecto básico que diferencia una arquitectura de otra es el método de descomposición del agente en tareas particulares.

La arquitectura es considerada como la manera en que se implementa el proceso de toma de decisiones de los agentes. Se pueden clasificar en cuatro tipos (Maes, 1994):

- Agentes basados en lógica

Estos agentes llevan a cabo el proceso de toma de decisiones mediante deducción lógica. El enfoque tradicional simbólico sugiere que, la conducta inteligente puede ser generada en un sistema con representación simbólica de su ambiente y de la conducta que se desea para él, a través de la manipulación sintáctica de dichas representaciones. La representación simbólica está constituida por fórmulas lógicas y la manipulación sintáctica se corresponde con la deducción lógica o la prueba de teoremas.

De manera general, los agentes implementados con esta arquitectura tienen elegancia y semántica limpia, aunque durante el proceso de toma de decisiones el ambiente no debe

cambiar y poseen alta complejidad computacional por lo que no operan bien en ambientes restringidos por el tiempo.

- Agentes reactivos

El proceso de toma de decisiones se realiza mediante un mapeo directo donde, dada una situación se obtiene la acción. Es un nuevo enfoque contrario al simbólico. Aquí los agentes construidos sí pueden operar en ambientes restringidos por el tiempo.

- Agentes con intereses, deseos y creencias

El proceso de toma de decisiones consiste en manipular estructuras de datos que son las que representan los intereses, deseos y creencias del agente.

- Capas

Se usan capas de software ordenadas de mayor a menor según la capacidad de razonamiento o nivel de abstracción.

Las investigaciones en el campo de los agentes virtuales inteligentes según (Dragone, Holz y Duffy, 2005) se dirigen hacia dos perspectivas diferentes:

1. Los aspectos físicos donde el objetivo principal es tratar de reproducir atributos físicos de agentes naturales o humanos, por ejemplo, agentes con apariencia humana (Maatman, Gratch y Marsella, 2005), (Kopp, Gesellensetter y Krämer, 2005), (Krämer, 2005).
2. Las conductas deliberativas y el modelado de usuarios.

Los mismos autores (Dragone et al., 2005) proponen trabajar en una tercera dirección: la frontera entre lo real y lo virtual con el objetivo de facilitar la integración de agentes físicos (en este caso un robot) y virtuales (una simulación de la actuación del robot) en interacción con los humanos.

1.1.2 Herramientas para la construcción de Agentes Virtuales Inteligentes (IVA⁹)

⁹ Siglas en inglés (Intelligent Virtual Agent)

El desarrollo de herramientas fáciles de usar es vital para la investigación gráfica. Una de las áreas más activas e interesantes de esta rama en los últimos años ha sido la de los agentes virtuales inteligentes.

Según (Gillies et al., 2005), suministrar herramientas amistosas para construir IVAs podría incrementar la comodidad en la producción de ambientes gráficos interactivos, reducir los costos e incrementar potencialmente su calidad.

Los IVAs tienen características que afectan el diseño de estas herramientas, la más importante de ellas es su conducta autónoma ya que depende del contexto en que se desarrolla el agente y el contexto puede incluir muchos aspectos; por ejemplo, la localización del IVA, la conducta de otros IVAs o la conducta de humanos presentes en el ambiente. El contexto, por tanto, consiste en una gran cantidad de información heterogénea. (Gillies et al., 2005) reducen a un número de variables continuas y discretas las características del contexto y proponen una herramienta que permite al usuario ver sucesivamente la conducta del agente en diferentes contextos, darle valor a las variables para modificar los contextos y editar la conducta del agente; si esta no es correcta entonces pasa al próximo contexto hasta obtener el resultado deseado.

1.1.3 Los Agentes Inteligentes en ambientes de enseñanza – aprendizaje

Los AIs han sido llevados al campo de la enseñanza de forma exitosa, dotando a los sistemas instruccionales de características distintivas de gran utilidad como el manejo de emociones, el apoyo efectivo del trabajo colaborativo y las interacciones pedagógicas adaptativas. De este modo es posible hablar de interfaces que pueden tomar la información y representarla con discurso, con gesto, con expresión facial, con postura, etc.

Los sistemas basados en AI pueden resultar ventajosos como medios de instrucción, entre otras razones, porque: (Julian y Botti, 2000)

- Los estudiantes pueden realizar preguntas en cualquier momento.
- El agente puede estar siempre “observando” la actuación del estudiante.

- El agente puede reconstruir y redefinir su actuación a partir de las acciones del estudiante.
- El agente puede adaptar su conducta a situaciones inesperadas.
- El estudiante puede tomar el control en cualquier momento.
- En caso de errores el agente puede ayudar a que el estudiante aprenda de ellos.
- En cualquier momento, el estudiante puede consultar en qué parte del proceso se encuentra y las tareas faltantes.

Aunque los agentes pedagógicos no están del todo explorados ni explotados, se pueden encontrar avances significativos en esta área. Un agente pedagógico es un agente que toma decisiones acerca de cómo mejorar el aprendizaje de un alumno, y el “entorno” que él observa en su proceso de aprendizaje. Para cumplir con sus metas, un agente pedagógico puede actuar como un tutor virtual, como un estudiante virtual, o como un compañero de aprendizaje que ayuda al estudiante. Un área de aplicación especialmente relevante de estos agentes es el entrenamiento de equipos de trabajo. Esta aplicación plantea retos interesantes como son el hecho de tener que tutelar simultáneamente a más de un estudiante, o el hecho de tener que manejar simultáneamente conocimiento “espacial” acerca del entorno y conocimiento “procedimental” acerca de las operaciones que pueden ser realizadas en el mismo (González, 2004).

Una de las ventajas de los sistemas tutoriales inteligentes sobre los “no inteligentes” es su capacidad de adaptarse a las habilidades, conocimientos y características personales de un estudiante en particular. Esta adaptabilidad se puede lograr a través del modelado de las propiedades del estudiante mediante un agente estudiante (Méndez y de Antonio, 2005).

COACH (Selker, 1994) es un tutor inteligente que proporciona asistencia personalizada a estudiantes utilizando la computadora para aprender sobre dominios específicos tales como el lenguaje de programación LISP o el Sistema Operativo Unix.

ADELE (Elliot, Rickel, y Lester, 1997) es un Agente para Ambientes de Aprendizaje a Distancia (Agent for Distance learning Environments) desarrollado en 1997. Es un agente pedagógico que se ejecuta en cada una de las computadoras e interacciona con cada uno de

los estudiantes cuando ellos navegan en los materiales del curso a través de la Web. ADELE es responsable de monitorear al estudiante y recoger sus acciones para adaptar la presentación del curso y reportar el comportamiento de cada uno de los estudiantes al servidor central al terminar la sesión.

PPP Persona (Plan basado en Presentaciones Personalizadas) fue creado en 1998 y es un agente pedagógico animado para presentaciones interactivas en la Web. La persona aparece en diferentes formas: figuras animadas o modelos 3D. EL PPP Persona sigue el aprendizaje a través de materiales Web usando presentaciones donde dibuja los elementos de las páginas Web y proporciona comentarios sintetizados (Peña, 2004).

CU ANIMATE desarrollado en el 2002, consiste en un conjunto de herramientas de software para habilitar conversaciones con caracteres animados. Los agentes animados están incorporados en un ambiente de aprendizaje interactivo libro-multimedia para ayudar a los niños a adquirir conocimiento a través de la lectura (Jiyong, Jie y Cole, 2002).

Otros avances en esta línea, son sistemas que animan a los niños a desarrollar y contar sus historias, capacitándolos para representar pensamientos simbólicamente y compartirlos con otros niños en su propio lenguaje. Uno de estos desarrollos, el “Storytelling” es un sistema que apoya el trabajo en forma colaborativa entre niños, proporcionando un ambiente para que estos aprendan las habilidades más importantes del lenguaje. El agente conversacional SAM es otro resultado importante, puede contar historias en forma colaborativa con los niños, está pensado para ser un niño más del grupo (Villarreal, 2006).

Por su parte (Choua, 2002), se refiere al desarrollo de agentes para motivar a diferentes grupos, por ejemplo agentes que enseñan en procesos de aprendizaje sociales, agentes que son alborotadores e incluso proporcionan opiniones erróneas con la intención de entregar un elemento disonante entre lo que el estudiante sabe y lo que el agente sugiere, de manera que obligan al estudiante a buscar nueva información para reducir la disonancia.

Las aplicaciones desarrolladas aprovechan las fortalezas de los agentes y brindan solución a problemas muy complejos donde las técnicas existentes son muy difíciles de implementar y/o mantener. Por otro lado, muestran cuánto se ha evolucionado desde la perspectiva de la concepción de crear sistemas donde se contemplaba solamente la interacción del alumno

con la computadora hasta considerar, en los nuevos sistemas, aspectos tales como: el entorno social que afecta al aprendizaje del alumno (González, 2004), así como las características del estudiante y sus conocimientos previos.

Los sistemas hipertextos adaptativos (SHA) son otro caso particular de utilización de los agentes inteligentes en sistemas educacionales. Se distinguen por la posibilidad de individualizar el acceso a la información de manera que esta se acomode y responda a la diversidad y necesidades de los usuarios, o sea, es la información la que se ajusta a las necesidades del alumno y no al revés como suele ocurrir.

1.1.3.1 Métodos y Técnicas de adaptación

En (Brusilovsky, 1996) se describen varios métodos y técnicas de adaptación. Un método es una generalización de una o varias técnicas de adaptación, es decir, un mismo método puede implementarse con varias técnicas. Estas, en cambio, son procedimientos para lograr que un SHA se adapte. A continuación se detallan los métodos y técnicas correspondientes a la presentación adaptativa y las que sirven de soporte a la navegación adaptativa.

Presentación Adaptativa

La presentación adaptativa se centra en mostrar la información de diferentes maneras, y reordena la información según el modelo de usuario (De Bra, Brusilovsky y Hoben, 1999). Para cubrir estos objetivos se divide en los siguientes métodos:

- a) Explicaciones prerequisites: Inserta definiciones de todos los conceptos presentes en los prerequisites que son desconocidos por el usuario, ejecutándolos antes de presentar una explicación.
- b) Explicaciones comparativas: Proporciona explicaciones de semejanza y diferencia entre conceptos cuando se conoce un concepto similar al presentado.
- c) Explicación de variantes: Almacena diferentes versiones de algunas partes del contenido para mostrar, a cada sujeto, la más apropiada según su modelo de usuario.
- d) Ordenación: Coloca descendientemente, según su grado de relevancia, la información más adecuada para los conocimientos y características del usuario.

Las técnicas de la presentación adaptativa se encargan de manipular el contenido para adecuarlo a las características del usuario. La mayoría de las técnicas se usan para adecuar el texto, pero también se pueden emplear en contenidos multimedia (Berlanga y García, 2004):

- a) Texto expansible: Cuando se selecciona una palabra marcada, esta se reemplaza por texto relacionado, contrayendo o expandiendo así la información que se presenta al usuario. La idea es evitar abrumar al usuario con información que no necesita, dejando que elija si desea o no tener más explicaciones sobre un tema en particular.
- b) Texto condicional: Toda la información sobre un concepto se divide en partes, cada una de las cuales se asocia a una condición relacionada con el nivel de conocimiento del usuario. Cuando se presenta la información, el sistema muestra sólo aquella donde la condición es verdadera.
- c) Variantes de páginas o fragmentos:
 - Variante de páginas: El sistema tiene dos o más versiones de la misma página y las presenta de acuerdo a las necesidades del usuario.
 - Variantes de fragmentos: Una página tiene varios conceptos, el sistema almacena versiones de explicación para cada concepto y el usuario obtiene aquellas que se ajustan mejor al conocimiento que posee sobre cada concepto.
- d) Técnica basada en marcos: Toda la información sobre un concepto se almacena en forma de marcos (frames). Cada marco tiene varias explicaciones, ejemplos, referencias, etc. Se ejecutan reglas para seleccionar qué marco es más apropiado presentar considerando las características del usuario.

Soporte a la navegación adaptativa

Los métodos de la navegación adaptativa se encargan de ayudar a los sujetos cuando estos navegan por el hiperespacio, guiándolos, orientándolos o facilitándoles la posibilidad de crear vistas personalizadas a sus intereses:

- a) Guía global: Se utiliza cuando los usuarios tienen un objetivo de información que se encuentra en uno o varios nodos de la Web, y necesitan navegar para encontrar la información.
- b) Guía local: Ayuda al usuario a realizar la navegación sugiriéndole los enlaces más relevantes a seguir desde el nodo actual.
- c) Orientación global: Su objetivo es auxiliar al usuario en su conocimiento sobre la estructura de todo el hiperespacio y su posición absoluta en él.
- d) Orientación local: Su fin no es guiar directamente al usuario sino, más bien, auxiliarlo a entender cuáles son los enlaces próximos y a decidir correctamente qué camino seguir.
- e) Gestión de vistas personalizadas: Organiza un espacio de trabajo electrónico para los usuarios que necesitan acceso a una parte más pequeña del hiperespacio. En este caso, los usuarios son los responsables de crear sus propias vistas de información.

Las técnicas del soporte a la navegación adaptativa manipulan los enlaces que contienen las páginas Web para representar información relevante y apropiada para cada sujeto. Así, existen diferentes maneras de manipular los enlaces; los propuestos por (Brusilovsky, 1996) y (Berlanga y García, 2004) se explican seguidamente.

- a) Guía directa: El sistema decide cuál es el mejor enlace que el usuario debe seguir y se lo presenta como única opción de navegación en forma de enlace o botón. Es la tecnología más simple, sin embargo, restringe por completo la navegación libre.
- b) Ordenación adaptativa de enlaces: Ordena, por relevancia, los enlaces de una página considerando el modelo de usuario y alguna característica de valor.
- c) Ocultamiento adaptativo de enlaces: Esconde, deshabilita o elimina los enlaces que no sean relevantes para el usuario, cuando no son significativos para sus objetivos o presentan información que no está preparado para comprender.
- d) Anotación adaptativa de enlaces: Marca o señala los enlaces para darle al usuario claves sobre su contenido, estado actual o relevancia. Algunos ejemplos de esta técnica son la anotación adaptativa booleana (visitado-no visitado; recomendado-no

- recomendado) o la metáfora del semáforo en donde cada color (rojo, amarillo, verde) representa el grado de conveniencia que tiene el enlace para el usuario.
- e) Generación adaptativa de enlaces: Este tipo de adaptación incluye tres casos: descubrir nuevos enlaces útiles entre documentos y agregarlos permanentemente al conjunto de enlaces existente, generar enlaces para navegación basados en la similitud entre elementos y recomendaciones dinámicas de enlaces relevantes.
 - f) Mapas adaptativos: Modifica la estructura de mapas hipermediales para cada individuo según su modelo de usuario.
 - g) Navegación pasiva: Consiste en agregar enlaces no-explicitos (sin texto de enlace) para ofrecer ayuda al usuario cuando se detecta un patrón de conducta, por ejemplo, que el usuario permanece inactivo durante cierto período de tiempo.

Tal y como ocurre en la vida real, muchos problemas en el mundo virtual no pueden ser resueltos por un solo agente o quizás pudieran ser mejor resueltos por varios de ellos. Surge así la necesidad de crear grupos de agentes que trabajen por un objetivo común, ocupándose cada uno de una parte de la tarea principal.

1.2 Sistemas Multiagentes (SMA).

Según (Peña, 2004) el concepto de sistema multiagente surge bajo la influencia del trabajo realizado por Marvin Minsky titulado: “The society of mind” (Minsky, 1986), donde un sistema complejo puede ser visto como un grupo de agentes simples especializados en un dominio concreto.

Demazeau define un sistema multiagente como un grupo de agentes organizados que interactúan en un ambiente común y tienen los siguientes elementos fundamentales (Demazeau, 2000):

- Agentes (Arquitectura interna de las entidades a procesar).
- Ambiente (Dominio de los elementos donde se producirán las interacciones entre las entidades).
- Interacciones (Elementos para construir interacciones internas entre entidades).

- Organización (Elementos para estructurar grupos de entidades de acuerdo a su función en el sistema multiagente).

Los efectos de la interacción de un agente X sobre un agente Y pueden ser favorables, neutros o perjudiciales para los objetivos del agente Y, por lo que en un sistema multiagente es preciso prever las acciones de unos agentes sobre otros. Para influir sobre lo que otro agente hará es necesario que exista comunicación entre ellos. Según (Fernández, 2005) los modelos de comunicación entre agentes son:

1. Modelo del pizarrón.

La pizarra es una estructura de datos que se utiliza como área de trabajo común a los agentes para poder intercambiar información, datos y conocimiento. Al igual que las pizarras de las aulas, son dispositivos para compartir información, con múltiples escritores y múltiples lectores. En este modelo no hay comunicación directa entre los agentes.

2. Modelo Actor.

Este modelo parte de un enfoque semántico para modelar objetos con un comportamiento dinámico, independiente e interactivo, a los cuales se les llama actores. Los actores se comunican entre sí a través de mensajes. Cuando un actor recibe un mensaje, puede cambiar su estado local, crear un nuevo actor o transmitir un mensaje. A cada actor se le asocia una zona de memoria conocida como dirección postal, donde se reciben los mensajes que le envían otros actores. Un actor también tiene un comportamiento que determina las acciones que puede ejecutar en respuesta a un mensaje.

3. Modelo de red de Contrato.

Proporciona una solución al problema de descomposición basado en la negociación entre agentes inteligentes autónomos. Propone que un agente (administrador) identifique la tarea a realizar y transmita un aviso a todos los agentes disponibles (contratistas potenciales), estos examinan la tarea y determinan una conducta a seguir. El administrador entonces, compara todas las conductas propuestas y concede el contrato al agente que considere más conveniente. Entonces se establece un canal de comunicación entre el administrador y el contratista (González y Barreiro, 1993).

4. Modelo del Facilitador.

El facilitador es una pieza de software que incorpora el conocimiento necesario para realizar un control inteligente y monitorear cada procedimiento de cooperación. No necesita conocer sobre el problema, su función es sólo organizar a los agentes (Cuenca y García, 1992).

1.2.1 Descripción de los sistemas multiagentes

La forma en que se describen las particularidades de un sistema multiagente ha sido preocupación de autores como (Bauer y Müller, 2004) y (Jonker, Klusch y Treur, 2000). Este último plantea que se necesitan lenguajes diferentes para especificar, por un lado, la arquitectura del sistema multiagente y por otro, las propiedades de tales agentes, de hecho propone un lenguaje para especificar las descripciones del diseño (DESIRE) y un lenguaje para especificar la conducta, los requerimientos y escenarios (TRL). A la hora de concebir un sistema multiagente la decisión más importante a tomar es determinar qué agentes el sistema contendrá y cómo ellos interactuarán (Bauer y Müller, 2004).

Se han creado diferentes metodologías con el fin de ayudar a los programadores, sobre todo, en las fases de análisis y diseño. MESSAGE¹⁰, Tropos¹¹, Prometheus¹², MaSE¹³, PASSI¹⁴ y AORML¹⁵ son algunas de ellas basadas en el lenguaje unificado de modelado (UML¹⁶). Por su parte (Alonso, 2005) hace un análisis interesante fundamentado en el resultado de investigaciones en el campo de la Ingeniería de Software Orientada a Agentes donde concluye que una metodología de desarrollo orientada a agentes debe tener las siguientes características:

1. No debe estar condicionada al uso del paradigma de agentes desde el análisis.
2. Debe conducir naturalmente a la conclusión de si es o no factible el desarrollo de un SMA.

¹⁰ www.springerlink.com/index/n314747k87876827.pdf

¹¹ www.troposproject.org/papers_files/aose02-23.pdf

¹² www.jamesodell.com/AOSE02-papers/aose02-04.pdf

¹³ www.springerlink.com/index/r86t3251838u3418.pdf

¹⁴ www.springerlink.com/index/ekchgfyn03fu5g.pdf

¹⁵ www.inf.ufes.br/~rguizzardi/publications/kimas05_guizzardi.pdf

¹⁶ <http://www.omg.org/technology/documents/formal/uml.htm>

3. Si es factible, entonces debe identificar sistemáticamente los componentes del MAS.
4. Debe conducir a un modelo organizacional.
5. Debe producir agentes reusables.
6. Debe ser fácil de usar y no requerir muchos conocimientos sobre la tecnología de agentes.

Se comparte el criterio de este autor, el que posteriormente propone la metodología SONIA que define un modelo de análisis genérico, independientemente de la arquitectura, incluyendo como primera etapa de la fase de diseño, una de síntesis que suaviza el paso del modelo conceptual al modelo formal.

1.2.2 Sistemas Multiagentes Inteligentes

La inteligencia artificial distribuida integra los conceptos de dos campos de conocimiento: la IA y los sistemas distribuidos. El campo de la IA intenta comprender las entidades inteligentes, pero, a diferencia de la filosofía y la psicología que también intentan hacer lo mismo, la IA intenta construir entidades inteligentes (Russell y Norvig, 1995). El campo de los sistemas distribuidos estudia las propiedades de conjuntos de procesadores autónomos que no comparten memoria primaria, pero sí cooperan comunicándose por medio del envío de mensajes sobre una red de comunicación (Silberschatz, Baer y Gagne, 2005).

A partir de los conceptos de IA y de sistema distribuido expuestos, la Inteligencia Artificial Distribuida (IAD) puede ser definida como un campo del conocimiento que estudia e intenta construir conjuntos de entidades autónomas e inteligentes que cooperan para desarrollar un trabajo y se comunican por medio de mecanismos basados en el envío y recepción de mensajes.

Desde que surgió la IAD se ha interesado, entre otros temas, por estudiar el modelo y comportamiento de varios agentes que cooperan entre sí para la resolución de un problema o desarrollo de una tarea. Los tres ejes fundamentales que se han estudiado en IAD son:

- Los SMA: estudian el comportamiento de agentes inteligentes que resuelven un problema de manera cooperativa.

- Resolución Distribuida de Problemas (RDP): trabaja con las formas de dividir un problema, para asignar las partes a un conjunto de entidades independientes y cooperantes, para que en grupo hallen la solución.
- La Inteligencia Artificial en Paralelo (IAP): se centra en el desarrollo de lenguajes y algoritmos paralelos para sistemas concurrentes en IAD.

Desde la perspectiva de la IAD un sistema multiagente es una red de entidades capaces de resolver un problema trabajando conjuntamente para encontrar respuesta a problemas que están más allá de la capacidad y del conocimiento individual de cada entidad. Así, en medios multiagentes, los agentes deben coordinar sus acciones y comunicar el conocimiento y la información que poseen (Ibarra, 2008).

Herramientas para el desarrollo de agentes.

En la misma medida en que el diseño orientado a agentes se ha ido desarrollando, han ido apareciendo herramientas de software para la construcción de sistemas basados en agentes. En este sentido, es posible hablar desde lenguajes para la programación de agentes (lenguajes para la programación de la estructura del agente, lenguajes de comunicación de agentes y lenguajes de programación del comportamiento del agente) hasta programación orientada a agentes. La idea clave de este paradigma es la programación directa de agentes. El paquete LPA Prolog (Shalfield, 2005) incluye el sistema Chimera¹⁷ que está especialmente concebido para construir sistemas multiagentes móviles que utilizan el paradigma de la Programación Lógica.

Por otro lado, las tecnologías más novedosas están en el campo del procesamiento y presentación de la información y no existe la información si el agente no es capaz de entenderla, por eso se están definiendo estándares para intercambio de información que permitan la interoperación entre agentes (García, 2007).

El ACL (Agent Communication Language) es un lenguaje de comunicación de agentes desarrollado por ARPA, que tiene tres componentes: un vocabulario, un lenguaje de contenido llamado formato de intercambio de información KIF (Knowledge Interchange Format) y un lenguaje de comunicación llamado KQML (Knowledge Query Manipulation

¹⁷ <http://www.cgl.ucsf.edu/chimera/docindex.html>

Language). Un mensaje de ACL es un mensaje en KQML que consiste de una directiva de comunicación y de un contenido semántico en KIF expresado en términos del vocabulario (Baray y Wagner, 1999) .

Para que esa información afecte al estado interno y a las acciones del agente en el futuro, el agente debe ser capaz de adaptarse. Las redes neuronales son una tecnología que permite asociar entradas a salidas, y adaptarlas para dar las salidas deseadas. También, la lógica difusa permite resumir el conocimiento en una serie de reglas que expresan la información con los mismos valores lingüísticos que lo hacen los seres humanos.

Por otro lado, los grupos de agentes pueden también evolucionar usando algoritmos genéticos, que podrían actuar sobre agentes de la misma forma que la evolución: eliminando los peores y dejando que los mejores se reproduzcan, cambiando a su vez con la posibilidad de crear algo mejor (García, 2007).

1.2.3 Sistemas Multiagentes en la Enseñanza

Existen variadas áreas de aplicación de los sistemas multiagentes: la robótica (Busquets, 2008), (Innocenti, 2008), (Ibarra, 2008), la medicina y la enseñanza son algunas de ellas. (Méndez y de Antonio, 2005) se refieren al proyecto MAEVIF (Model for the Application of Intelligent Virtual Environments to Education) que tiene como objetivo definir un modelo para la aplicación de ambientes virtuales inteligentes en la educación el cual incluye la definición de un modelo genérico para ambientes de aprendizaje inteligentes basados en el uso del mundo virtual, la definición de una arquitectura de software basada en agentes para apoyar el modelo genérico, el diseño e implementación de un prototipo de herramienta de autor que facilite la creación de dichos ambientes y la definición de un conjunto de recomendaciones metodológicas para su desarrollo.

Por su parte (Ovalle y Jiménez, 2005), plantean que los principios de los MAS han mostrado un potencial adecuado en el desarrollo de sistemas de enseñanza donde:

- La naturaleza de los problemas de enseñanza-aprendizaje son más fácilmente resueltos a través de un abordaje cooperativo (Oliveira y Viccari, 1996).

- Los sistemas deben ser capaces de adaptarse a los cambios en la estructura del ambiente (Ferber, 1999).
- Los agentes de software pueden representar conocimiento pedagógico y/o desempeñar tareas de tutoría para soportar y facilitar el aprendizaje humano (Giraffa y Viccari, 1998).

El Ambiente de Enseñanza Aprendizaje multiagente ALLEGRO está conformado por un Sistema Tutorial Inteligente que permite brindar aprendizaje en forma individualizada y un Ambiente Colaborativo. El ambiente fue modelado a través del enfoque de un MAS debido a que ofrece las cualidades de autonomía, flexibilidad y adaptabilidad (Ovalle y Jiménez, 2005).

Cuando se habla de autonomía se refiere a la iniciativa que toman los agentes del MAS para realizar acciones pedagógicas sin la intervención de los humanos con el propósito de que el aprendiz logre los objetivos. Cuando se habla de flexibilidad se refiere a que el aprendiz no sólo utiliza una metodología para su aprendizaje (Individualizada/Colaborativa). Cuando se habla de adaptabilidad se refiere a que el contenido de la instrucción se moldea de acuerdo a las necesidades específicas y preferencias del aprendiz.

El proyecto “Negociación Pedagógica en ambiente AMPLIA” (Fernández, 2005) del Instituto de Informática de la Universidad Federal de Río Grande del Sur (Porto Alegre) tiene como objetivo la construcción de un ambiente de enseñanza-aprendizaje computacional basado en Web multiagente, que se usará como un recurso adicional para la formación de estudiantes de medicina.

De forma general, el alumno modela un diagnóstico y su modelo es comparado con el formulado por especialistas y forma parte del conocimiento del sistema. Las diferencias entre los modelos se tratan de acuerdo a estrategias pedagógicas basadas en la interacción y la negociación entre los agentes internos del sistema y los alumnos.

En (Peña, 2004) se describe cómo suministrar características adaptativas a un sistema manejador de cursos diseñado para e-learning usando tecnología de agentes. MACES, BAGHERA, MAS-PLANG y JADE son otras arquitecturas para sistemas de enseñanza también referidas por (Ovalle y Jiménez, 2005).

En el presente trabajo se conjugan tres elementos esenciales: el primero y más significativo es la Programación Lógica como objeto de estudio, el segundo es el uso de agentes inteligentes que permiten la adaptación de los contenidos al estado cognitivo del educando y el tercero es el uso de los mapas conceptuales como un medio de navegación dentro de las entidades.

1.3 Los mapas conceptuales

Los mapas conceptuales iniciaron su desarrollo, durante la década de los 70, en el Departamento de Educación de la Universidad de Cornell, EE UU y se basan en la teoría del aprendizaje significativo de Ausbel, que plantea que hay aprendizaje significativo cuando se intenta dar sentido o establecer relaciones entre los nuevos conceptos o nueva información, los conceptos y conocimientos existentes ya en el alumno, o con alguna experiencia; es decir, hay aprendizaje significativo cuando la nueva información puede relacionarse de modo no arbitrario y sustancial con lo que el alumno ya sabe (Ontoria, 1993).

La idea de organizar contenidos en forma de mapa conceptual fue concebida por Novak (Novak y Gowin, 1988), como una técnica que representa, simultáneamente, una estrategia de aprendizaje, un método para captar lo más significativo de un tema y un recurso esquemático para representar un conjunto de significados conceptuales incluidos en la estructura de proposiciones. Los elementos básicos que lo componen son: los conceptos, las palabras-enlace y las proposiciones. En la figura 1.1 se muestra un mapa conceptual donde los conceptos Programa, Predicados, Preguntas y Comentarios están relacionados por las palabras enlace “es un conjunto de”, se consulta a través de” y “puede contener” formando proposiciones que se leen: un programa es un conjunto de predicados, un programa se consulta a través de preguntas y un programa puede contener comentarios.

Una forma más gráfica de definir el mapa conceptual y vincularlo con el aprendizaje significativo es considerarlo, en cierto modo, homogéneo a los mapas de carreteras, donde los conceptos representan las ciudades y las proposiciones las carreteras que las enlazan.

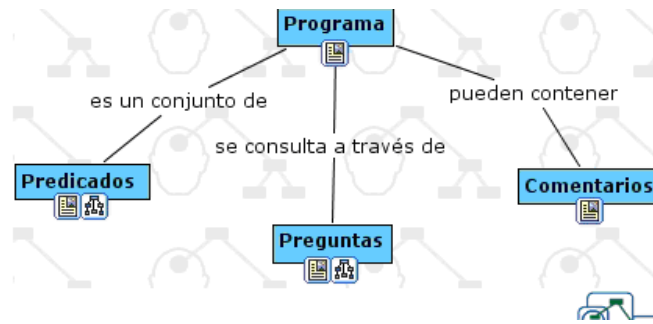


Figura 1.1 Mapa conceptual sobre el concepto “programa Prolog”.

En la literatura revisada aparecen diversas clasificaciones de los Mapas Conceptuales. (Vizcarro y León, 1998) los dividen en las siguientes categorías:

- Mapa araña: Está organizado de forma que el concepto central o factor unificador se coloca en el centro del mapa y los subtemas se distribuyen a su alrededor.
- Mapa jerárquico: Presenta la información en un orden descendente de importancia, de forma que la información más relevante se encuentra en el nivel superior.
- Diagrama de flujo: Organiza la información en un formato lineal.
- Mapa de sistemas: Organiza la información en un formato similar al diagrama de flujo, con la salvedad de que incluye entradas y salidas.

Se puede afirmar que no existe un mapa conceptual correcto y único para un tema dado. De acuerdo con el nivel de profundidad del asunto a tratar, su complejidad puede variar desde una concepción muy simple hasta una en la que existan cientos de conceptos y palabras-enlace. En última instancia, su elaboración depende de los conceptos previos que tenga el individuo que lo elabora, así como del nivel de jerarquización que sea capaz de darle según la importancia que establezca entre los conceptos y el nivel de inclusión que perciba de ellos. Lo importante son las relaciones que se establecen entre los conceptos a través de las palabras-enlace para formar proposiciones que configuran un valor de verdad sobre el objeto estudiado (Estrada y Flebes, 2002).

1.3.1 Los mapas conceptuales en la enseñanza

Los mapas conceptuales se utilizan en medida creciente en todas las actividades en las que es necesario representar, utilizar y gestionar el conocimiento, sea en ambientes

empresariales o en ambientes formativos. Simular el modo en que el conocimiento se conserva y se recupera en la mente humana confiere a esta técnica un lugar indiscutible entre las estrategias más actuales para estimular el aprendizaje activo.

Esta técnica resulta muy útil en diversas etapas del proceso educativo. Según (Dávila y Martínez, 2000) se pueden usar en la planeación, como recurso para organizar y visualizar el plan de trabajo, en el desarrollo, como una herramienta que ayuda a los estudiantes a captar el significado de los materiales que pretenden aprender y como recurso para la evaluación formativa permitiendo “visualizar el pensamiento del alumno”, para así corregir a tiempo, posibles errores en la relación de los conceptos principales.

Respecto a las destrezas cognitivas, Estrada y Febles apuntan en (Estrada y Flebes, 2002) que los mapas conceptuales desarrollan:

- Las conexiones con ideas previas, tanto en su confección antes del desarrollo del tema como en su tratamiento posterior.
- Capacidad de inclusión, dada la jerarquización de los conceptos y el nivel de comprensión que implica su relación.
- La diferenciación progresiva entre conceptos, sobre todo si se elaboran en diferentes momentos del desarrollo del tema.
- La integración o asimilación de nuevas relaciones cruzadas entre conceptos.

También señalan que desde el punto de vista del colectivo contribuye a lograr una mayor uniformidad en el nivel de conocimiento de los integrantes del grupo.

Los mapas conceptuales hechos por el profesor incrementan el aprendizaje y la retención de información y permiten que los estudiantes tengan un modelo conceptual claro de la tarea, que sirva de guía para actuar y mejorar sus habilidades de autoevaluación. Aún cuando la persona que aprende solo sea capaz de realizar una parte de la tarea, el hecho de tener un modelo conceptual claro del conjunto de actividades le ayuda a ver el sentido de la parte que está realizando (Vizcarro y León, 1998).

Existen diferentes aspectos que el docente debe tener presente cuando va a evaluar un mapa conceptual, entre los que se encuentran la correcta jerarquía de los conceptos, la cantidad y

calidad, la buena relación de los significados entre dos conceptos conectados y que exista una conexión significativa entre un segmento de la jerarquía y el otro.

La organización de la información es una de las principales aplicaciones de los mapas conceptuales. Como herramienta gráfica, los mapas conceptuales permiten presentar, de manera esquemática, las unidades de información relevante contenidas en un documento, las relaciones entre dichas unidades, y entre estas y las otras fuentes de información. Así, una vez construidos los mapas de las diferentes fuentes, el estudiante puede encontrar, de una manera visual, las posiciones conceptuales en conflicto, las posiciones conceptuales complementarias y las posiciones ambiguas.

1.3.2 Los mapas conceptuales y las tecnologías de la información

Las tecnologías de la información han potenciado los mapas conceptuales. Esta interacción ha dado lugar a espacios virtuales de conocimiento que permiten al alumno operar sobre ideas, representaciones de la información, evaluar sus niveles previos de comprensión y llegar a ser conscientes de su propio aprendizaje. Simón Cuevas, en su artículo “Propuesta de aplicación de los mapas conceptuales en un modelo pedagógico semipresencial” (Cuevas, 2007), plantea que en los últimos años han alcanzado una gran popularidad y una gran integración con las tecnologías computacionales y de las comunicaciones, se han convertido en un elemento muy importante en los planes de perfeccionamiento de los sistemas de enseñanza y han extendido su uso a otras esferas de la actividad humana en las que la gestión del conocimiento ocupa un lugar preponderante. Un ejemplo de ello es el uso en la educación a distancia donde es fundamental el estudio independiente y el autoaprendizaje por parte de los estudiantes.

Teniendo en cuenta que los mapas conceptuales constituyen una representación explícita y manifiesta de los conceptos y proposiciones que posee una persona, se puede decir que facilitan el intercambio de puntos de vista entre profesores y alumnos acerca de la validez de un vínculo proposicional determinado e incluso estos últimos se pueden percatar de las conexiones que faltan entre los conceptos, por esta razón, esta técnica ha articulado tan favorablemente con los preceptos de la educación a distancia que supone que estudiantes y profesores no estén físicamente en el mismo lugar y al mismo tiempo. Utilizando tanto la

vía sincrónica como asincrónica, se puede intercambiar desde lejos la interpretación que se tiene sobre un concepto, su definición, alcance y sus relaciones con otros conceptos en un contexto determinado (Cuevas, 2007).

Los mapas conceptuales como herramientas de navegación

Los sistemas de hipermedia, que la autora de esta tesis considera especialmente útiles para construir ambientes de enseñanza aprendizajes asistidos por computadora, se basan en el principio de la facilidad de acceso a grandes cuerpos de información. Sin embargo, existe la posibilidad de que los estudiantes se pierdan dentro de ellos al no contar con un índice, una tabla de contenido o estructura similar que los ayude a orientarse. Por otra parte al leer el documento se están tomando continuamente decisiones sobre qué enlaces seguir y cuáles ignorar, lo cual puede producir una sobrecarga en el sistema cognitivo y tener como consecuencia que se pierdan los objetivos de aprendizaje trazados inicialmente.

Los mapas conceptuales si se emplean como método o herramienta de diseño que sirven como técnica de andamiaje estructural, antes y durante el desarrollo de los productos de hipermedia ofrecen una solución a estos problemas. De este modo se convierten en un mecanismo idóneo de navegación para los estudiantes que necesitan cierta orientación mientras exploran dominios muy extensos de información (Vizcarro y León, 1998), al proporcionar una visión gráfica de los diferentes nodos y enlaces.

Sin embargo, la autora de este trabajo considera que para complementar los mapas y aumentar sus posibilidades como herramientas para la navegación, no basta con lo enunciado anteriormente y sostiene la idea de adicionar inteligencia a la navegación. La inteligencia mencionada se puede lograr usando diferentes técnicas de IA, en este caso particular se proponen los agentes inteligentes.

1.4 La enseñanza de la Programación Lógica

Prolog es el más extendido de los lenguajes de programación lógica (del francés Programation et Logique), fue ideado a principios de los años 70 en la universidad de Aix-Marseille por los profesores [Alain Colmerauer](#) y Phillipe Roussel, se basa en el cálculo de predicados de primer orden y se utiliza ampliamente en investigaciones de IA, lo que

justifica su inclusión en los programas de estudios de las carreras Ciencia de la Computación e Ingeniería Informática.

Prolog (Bratko, 1986) es un lenguaje descriptivo, esto significa que el programador sólo necesita proporcionar una descripción del problema y plantear las reglas para resolverlo. A partir de allí, Prolog determina cómo encontrar la solución. Esta filosofía contrasta con la de otros lenguajes de computadora tradicionales, en los que el programador debe proporcionar instrucciones que dicen exactamente y paso a paso cómo resolver un problema dado. En otras palabras, el programador debe saber resolver el problema antes de que la computadora pueda hacerlo.

Los planes de estudios vigentes en Cuba sitúan la enseñanza del Prolog después que los estudiantes conocen el paradigma imperativo. En específico, en el plan de estudio C para la carrera de Ingeniería Informática, la Programación Lógica forma parte del sistema de contenidos de la asignatura Programación Descriptiva (que se imparte en el cuarto año) la que tiene como objetivo preparar a los estudiantes para que, conociendo los fundamentos generales y evolución de los lenguajes de programación descriptivos, sean capaces de modelar problemas utilizando paradigmas funcionales y lógicos, hacer programas utilizando algunas de las versiones de los lenguajes descriptivos, así como evaluar, seleccionar y asesorar en la compra de ambientes de programación, medios técnicos y la asimilación de nuevas tecnologías a ser utilizados en las aplicaciones que utilicen la programación descriptiva.

Para ello, los estudiantes deben conocer los fundamentos, evolución y características de los lenguajes de programación descriptivos, las posibilidades y limitaciones de los lenguajes prescriptivos, los rasgos de la programación y lenguajes descriptivos (funcionales y lógicos), así como las estructuras de datos y la programación recursiva en estos lenguajes utilizando las estructuras de datos definidas. También deben estudiar las posibilidades y limitaciones del software y hardware relacionados con las versiones de los lenguajes descriptivos.

El plan de estudios D, para esta misma carrera, sitúa la Programación Lógica dentro de la asignatura Introducción a la Inteligencia Artificial que se imparte en segundo año. En la

carrera Ciencia de la Computación, el plan D ubica la Programación Lógica dentro de la asignatura Programación Descriptiva que los estudiantes reciben en el cuarto año de la carrera.

Todo lo que se ha referido anteriormente, transforma la enseñanza del lenguaje en un gran reto para los profesores debido a que este paradigma de programación difiere, en gran medida, de los paradigmas estudiados con anterioridad por los educandos, los cuales, muchas veces, pretenden extrapolar las ideas de los paradigmas conocidos.

Estudios comparativos entre los estilos imperativos y declarativos han revelado que las dificultades de los estudiantes en la utilización de Prolog provienen de diseñar la solución al problema en términos de estructuras propias del estilo procedimental, tales como: if, while, repeat, etc., y no encontrar, al momento de la implementación, primitivas Prolog para traducir dichas estructuras (González, 2007).

En opinión de (Lezcano y Valdés, 1998) la enseñanza de la Programación debe comenzar por la forma declarativa y relatan experiencias propias en la enseñanza de la Programación Lógica a niños pequeños con resultados sorprendentes. Esto se debe a que la forma de pensar del ser humano se acerca más a la lógica y cuando se enseña a programar en forma imperativa realmente se trata de imponer un razonamiento que no es propio de los humanos y que está determinado, en última instancia, por la arquitectura secuencial de las computadoras.

1.4.1 La enseñanza de la Programación Lógica en Cuba

En la década del 90, un grupo de profesores de la UCLV, liderados por el Dr. Mateo Lezcano Brito comenzó un estudio sobre la enseñanza de la Programación Lógica. Algunos de los primeros resultados se recogen en el informe de su tesis de doctorado (Lezcano, 1998). La experiencia, en ascenso, de sus fundadores ha potenciado el grupo, al que se han sumado profesores de otras universidades. En este trabajo se ha profundizado acerca de las dificultades identificadas por los autores anteriores y se han añadido otras.

- Dificultad 1. Los estudiantes no saben describir el problema planteado.

Uno de los problemas que se enfrentan al enseñar lenguajes de este tipo es que los estudiantes no son capaces de describir, en forma general, el problema que se plantea y por esa razón presentan soluciones incorrectas total o parcialmente incorrectas. Se toma un ejemplo para clarificar la idea, se pide a los estudiantes que resuelvan el siguiente problema:

“A partir del siguiente programa Prolog, escriba una interrogante para encontrar dos personas que tengan el mismo padre (parent).

```
parent(pam, bob) .
    parent(tom, bob) .
    parent(tom, liz) .
    parent(bob, ann) .
    parent(bob, pat) .
    parent(pat, jim) .

offspring( Y, X) :-
    parent( X, Y) .
```

Fragmento 1.1. Programa de relaciones familiares.

La “solución” de algún estudiante es escribir la interrogante de la forma siguiente: `parent(bob, X)`. Prolog tiene dos respuestas para la pregunta anterior: `X=ann` y `X=pat`.

Aparentemente, se ha respondido bien el ejercicio (según la apreciación del estudiante) porque la respuesta es correcta para el caso específico del padre llamado bob. Realmente la respuesta no es correcta ya que la pregunta debe hacerse en forma general.

Debe observarse que se están buscando dos personas que tengan el mismo padre y no dos personas que tengan el mismo padre y que su padre sea bob (ese es un caso particular). Se aprecia que el estudiante no es capaz de escribir una regla (en este caso una regla tipo interrogante) declarando exactamente, en forma general, lo que se pide.

Una “solución” que puede aceptarse en una etapa inicial del curso es la siguiente:

```
parent(X, Y) , parent(X, Z) .
```

Las respuestas de Prolog (obtenidas por alternativas) se pueden apreciar en la tabla 1.1.

Se puede observar que Prolog da diez “soluciones” al problema pero en realidad no todas son soluciones al problema enunciado y sí soluciones a la pregunta que se hizo (que no responde de forma exacta al enunciado del problema). Puede apreciarse la tabla 1.2 donde se detallan las categorías de las “soluciones”.

Tabla 1.1 Respuestas a la interrogante: parent(X, Y), parent(X, Z) al fragmento de programa 1.1			
Respuestas	Valores de las variables		
	X	Y	Z
1.	pam	bob	bob
2.	tom	bob	bob
3.	tom	bob	liz
4.	tom	liz	bob
5.	tom	liz	liz
6.	bob	ann	ann
7.	bob	ann	pat
8.	bob	pat	ann
9.	bob	pat	pat
10.	pat	jim	jim

Tabla 1.2. Respuestas, por categorías, a la interrogante: parent(X, Y), parent(X, Z) al fragmento de programa 1.1			
	a) Soluciones no válidas	b) Soluciones reales	c) Soluciones repetidas
	Variables	Variables	Variables

	X	Y	Z	X	Y	Z	X	Y	Z
1.	pam	bob	bob	tom	bob	liz	tom	liz	bob
2.	tom	bob	bob	bob	ann	pat	bob	pat	ann
3.	tom	liz	liz						
4.	bob	ann	ann						
5.	pat	jim	jim						
6.	bob	pat	pat						

La primera categoría (soluciones no válidas) surge de tomar los mismos valores para las variables Y y Z, dado un valor para la variable X. La segunda categoría (soluciones reales) es la solución correcta al problema enunciado. La tercera categoría (soluciones repetidas), está bien pero realmente es una alternativa de la segunda categoría intercambiando los valores de las variables Y y Z para cada valor de X correcto y por tanto es una repetición.

Una solución correcta, que no siempre la logran los estudiantes o que solo logran algunos estudiantes aventajados, es la siguiente: $\text{parent}(X, Y), \text{parent}(X, Z), Z \neq Y$.

En esta variante se fuerza a que el valor de la variable Z sea diferente al valor de la variable Y, de esta forma se evitan todas las soluciones de la variante a en la tabla 1.2 (son las no válidas).

En realidad el programa da también las soluciones repetidas que se aprecian en las secciones b y c de la tabla 1.2. Ese problema puede hacer pensar en el uso del operador de corte (!) para evitar las repeticiones y la interrogante anterior quedaría de la forma siguiente: $?\text{parent}(X, Y), \text{parent}(X, Z), Z \neq Y, !$.

El problema en este caso es que el corte inhibe el backtracking y por eso el programa solo dará una solución al problema cuando en realidad tiene dos (o más en otros casos).

Para obtener las soluciones sin estas últimas repeticiones habrá que modificar el programa y no la interrogante ya que la modificación a la interrogante puede requerir de muchos artificios y no tendría sentido hacerlo en un lenguaje que se distingue por su sencillez y la elegancia de las soluciones. El análisis de este problema es un ejercicio interesante que puede plantearse a los estudiantes.

- Dificultad 2. Escriben reglas con antecedentes (parte del cuerpo) no definidos (no son ni hechos ni conclusiones de otras reglas). Por ejemplo:

Se pide que describan la relación abuelo (grandparent) y lo hacen de la forma siguiente:

```
parent (pam, bob) .  
...  
parent (pat, jim) .  
grandparent (X, Y) :-  
    son (Y, Z) ,  
    son (Z, X) .
```

Fragmento 1.2. Programa de relaciones familiares.

En este caso se puede observar que se describe al predicado grandparent en función de un predicado son que no existe. El error puede considerarse doble, el primero es definir un predicado en función de uno que no existe y el segundo es que, tal vez, para buscar la solución al problema se definan los hechos bases son y se llene la base de conocimientos con hechos bases perdiendo de esta forma la potencia del lenguaje que permite inferir datos que no están almacenados explícitamente.

- Dificultad 3. No se expresa exactamente lo que se necesita a través de una relación. Por ejemplo se puede citar la siguiente situación: Se pide que describan la relación siguiente: “alguien tiene un hijo” y lo hacen de la forma siguiente:

```
child (Y, X) :-  
    parent (X, Y) .
```

Cuando debería ser:

```
hasChild (X) :-  
    parent (X, _) .
```

Lo cual le resta claridad y naturalidad a lo que se expresa

- Dificultad 4. Mal uso de los términos compuestos o estructuras.

Cuando se trabaja con una estructura dentro en un predicado como el ejemplo siguiente:

```
datosPersonales (Nombre, fecha (DD, MM, AA) , S) :-
```

```
persona(Nombre, fecha(DD, MM, AA), S).  
persona(juan, fecha(12, 10, 82), 800).  
...  
persona(luis, fecha(01, 09, 97), 500).
```

Pretenden definir un predicado `fecha` cuando en realidad `fecha` es un dato que forma parte del predicado `persona`.

- Dificultad 5. Abstractar la esencia de un problema y no extrapolar ideas.

No se logra abstraer la esencia de los problemas para representarlo en el paradigma de la Programación Declarativa, muchas veces ni con estudiantes destacados. Lo anterior da por resultado que el pensamiento no sea el adecuado y que se usen técnicas no propias de este paradigma que, aunque son válidas en otros paradigmas, en el de la programación declarativa producen programas ineficientes y difíciles de entender que no aprovechan los mecanismos fundamentales del paradigma.

- Dificultad 6. Definición de listas.
 - a) No se percibe adecuadamente la definición recursiva de lista y por ese motivo los programas no funcionan en forma correcta.
 - b) No se manejan de forma apropiada los patrones de listas perdiendo la potencia expresiva que tienen para definir muchos problemas.
 - c) No contemplar la generalidad de los problemas que se resuelven con listas muchos de los cuales se definen, básicamente, en un predicado con tres cláusulas:
 - La primera como caso particular que expresa la condición de terminación de un predicado recursivo a partir de cumplirse algo con relación a la lista (por ejemplo estar vacía).
 - La segunda como el caso en que se cumple la condición buscada en la cual hay que tomar la decisión de qué hacer con el elemento de lista encontrado (por ejemplo guardarlo en una lista que se está construyendo).

- La tercera como el caso que no cumple la condición buscada y en la cual hay que tomar la decisión de qué hacer con el elemento de la lista que no la cumple (por ejemplo desecharlo).

Al observar este análisis, que no pretende abarcar todas las complejidades de la enseñanza de esta disciplina, puede apreciarse la necesidad de perfeccionar los medios que ayudan a mejorar la calidad de un proceso de enseñanza-aprendizaje que tiene características propias, que resulta complejo y que puede estar viciado por técnicas aprendidas en otros paradigmas que nada tienen que ver con el paradigma de la programación lógica. Las nuevas herramientas deben tener incorporados los resultados de la presente investigación, permitir su uso en diferentes formas de enseñanza (presencial, semipresencial y a distancia) y deben, además, potenciar su uso en forma independiente o colaborativa.

1.5 Conclusiones Parciales

El capítulo ha hecho un estudio de tres aspectos básicos de la presente investigación: la enseñanza de la Programación Lógica, los agentes como vía para lograr adaptación y los mapas conceptuales como medio de organización y navegación, lo que permite concluir que:

1. Los mapas conceptuales constituyen una forma adecuada para organizar la información en ambientes de enseñanza-aprendizaje basados en la navegación.
2. La inteligencia colectiva lograda a partir del esfuerzo de agentes individuales hace de los sistemas multiagentes una herramienta capaz de manejar técnicas para lograr adaptación en ambientes de enseñanza-aprendizaje.
3. La enseñanza de la Programación Lógica no es una tarea fácil y precisa de medios que logren un mejor uso de las tecnologías actuales.

El capítulo siguiente describe un sistema multiagente que permite presentar contenidos de la Programación Lógica organizados sobre mapas conceptuales que se configuran dinámicamente gracias a la acción de AI que actúan de acuerdo a los conocimientos individuales de los estudiantes.

Capítulo 2

APA-Prolog: un ambiente de
enseñanza-aprendizaje asistido por
computadora para la Programación
Lógica

2. APA-PROLOG: UN AMBIENTE DE ENSEÑANZA-APRENDIZAJE ASISTIDO POR COMPUTADORA PARA LA PROGRAMACIÓN LÓGICA

Partiendo del reconocimiento de la complejidad de la enseñanza de la Programación Lógica, de las principales dificultades detectadas en el estudio realizado y las opiniones de expertos de la UCLV y del CUSS se desarrolló el Ambiente para la Enseñanza y el Aprendizaje de la Programación Lógica (APA-Prolog¹⁸). El sistema ha sido elaborado teniendo en cuenta premisas técnicas y pedagógicas que permitan mejorar el proceso de enseñanza-aprendizaje.

En este capítulo se describen:

1. Las características generales de APA-Prolog.
2. La arquitectura del sistema multiagente para APA-Prolog.
3. El conjunto de recursos informáticos disponibles en el ambiente.

2.1 Características del Ambiente de Enseñanza-Aprendizaje para la Programación Lógica (APA-Prolog)

Los mapas conceptuales se pueden usar para organizar el conocimiento de una manera adecuada gracias a su fuerza expresiva y a la facilidad con que los observadores captan el significado de lo que en ellos está representado. Basado en esas propiedades, la interfaz de APA-Prolog es un mapa conceptual que contiene la mayoría de los conceptos asociados a la Programación Lógica. Esta forma de organizar la información permite relacionar los conceptos y ofrece un medio cómodo de navegación.

La interfaz ofrece una vía de acceso a diferentes recursos informáticos, tales como: textos explicativos, simuladores, entrenadores, evaluadores, un espacio para el intercambio y un directorio temático, los cuales permiten profundizar teóricamente acerca de los conceptos, analizar ejemplos, comprender funcionamientos, intercambiar ideas, etc.

¹⁸ Disponible en: <http://200.45.155.145:8084/cussprolog/default.html>

La navegación por el sistema puede ser libre o asistida. En el primer caso, el usuario recorre los contenidos de acuerdo a sus deseos y no existe restricción alguna en relación al orden de precedencia con que se visiten los nodos del mapa.

La navegación asistida está controlada por un conjunto de agentes inteligentes que toman decisiones basadas en los conocimientos previos del estudiante. Los agentes deciden qué recursos activar en cada caso, basándose en el estado del conocimiento del estudiante (bien, regular o mal). La figura 2.1 ilustra como se relacionan todos los componentes durante la navegación asistida.

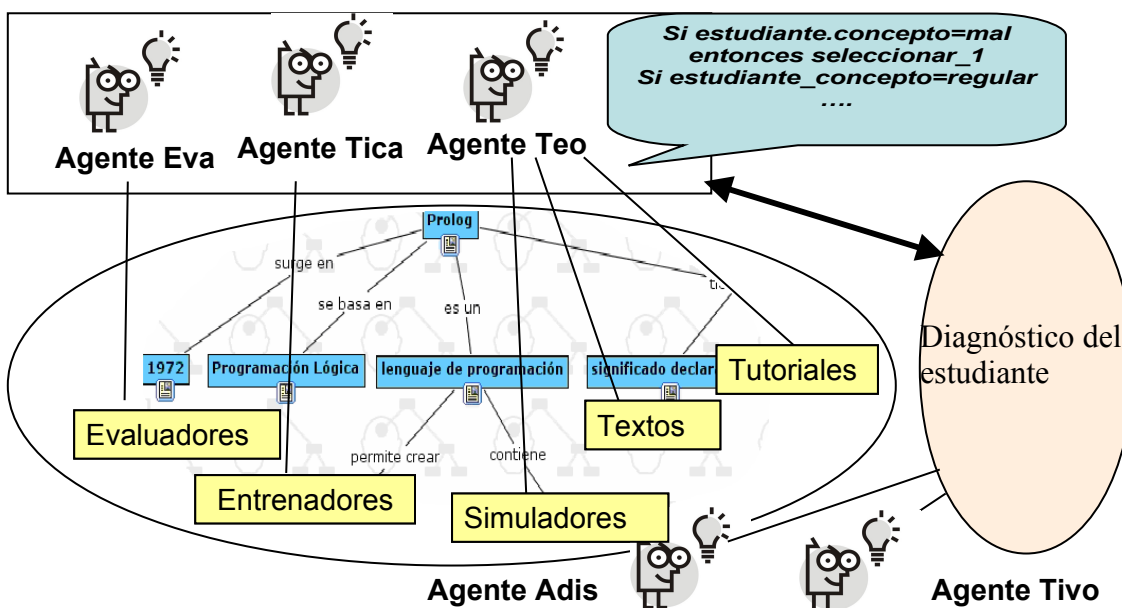


Figura 2.1 Interacción de todos los componentes durante la navegación asistida.

Para lograr la adaptación, el sistema realiza diferentes pasos. En el inicial determina el estado de conocimiento del estudiante acerca de cada concepto, una vez que se ha obtenido esa información se decide qué recursos presentar. La figura 2.2 muestra, en forma esquemática, la responsabilidad de cada uno de los agentes del sistema y la interacción entre los diferentes componentes. La figura 2.3 ilustra el proceso de adaptación del sistema a los conocimientos de un estudiante.

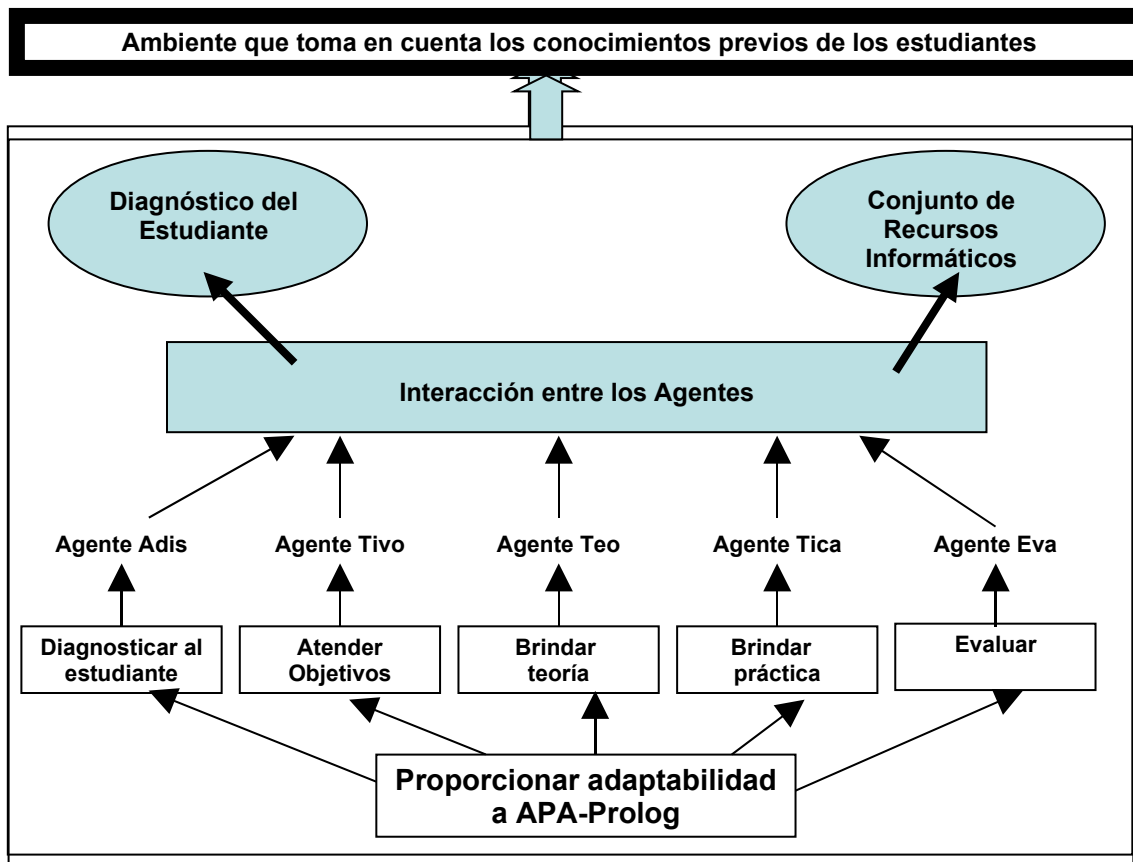


Figura 2.2 Interacción entre los componentes del sistema.

El desarrollo de APA-Prolog se trazó los siguientes objetivos:

1. Aprovechar las ventajas que brindan las computadoras y las TICs para crear ambientes virtuales de enseñanza-aprendizaje sin restricciones de espacio y tiempo.
2. Lograr un proceso de enseñanza-aprendizaje activo y significativo, protagonizado por el estudiante, facilitando los recursos que le permita convertir la información en conocimiento.
3. Obtener un ambiente que tome en cuenta los conocimientos previos del estudiante.
4. Apoyar el proceso de enseñanza-aprendizaje de la programación Lógica.

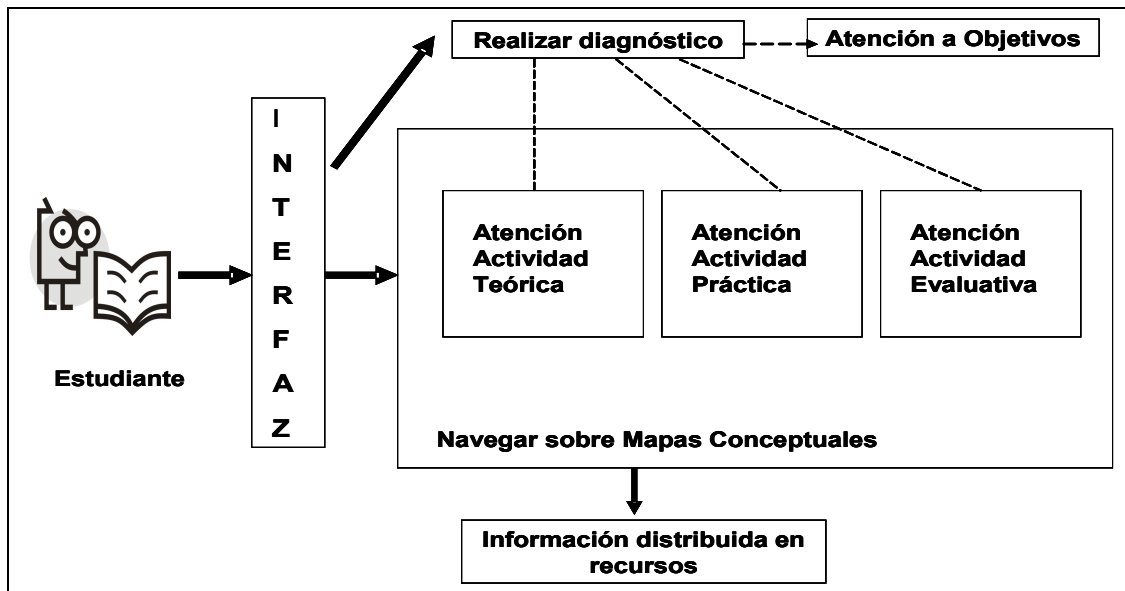


Figura 2.3 Proceso de adaptación de APA-Prolog.

2.2 Arquitectura básica de un agente

Como se explicó en el epígrafe anterior, cada uno de los agentes de APA-Prolog tiene una tarea específica que cumplir. Aplicando la técnica “Divide y Vencerás” cada una de ellas da lugar a otro grupo de actividades que los agentes deben realizar para conseguir un resultado exitoso. Estas actividades, de forma general, originan un conjunto de reglas que relacionan el estado de conocimiento del estudiante acerca del concepto estudiado con los recursos informáticos disponibles. La figura 2.4 muestra la arquitectura básica de un agente. Seguidamente se describen las características individuales de cada uno de ellos.

2.2.1 Agente Adis

La necesidad de crear ambientes computacionales capaces de mantener el interés de sus usuarios, implica el desarrollo de interfaces personalizadas para ofrecer un servicio que permita un trato individualizado al usuario en particular (Medina, 2007), adaptando la interacción con el sistema a necesidades e intereses personales. Para ofrecer un servicio adecuado a cada usuario es necesario que el sistema cuente con una representación de sus características propias, y en base a dicha representación, tome las decisiones pertinentes en

la interacción (Fischer, 2000). Para llevar a cabo las tareas anteriores, el sistema realiza un diagnóstico inicial a cada uno de los estudiantes que interactúan con el ambiente siendo esta la misión asignada al agente Adis.

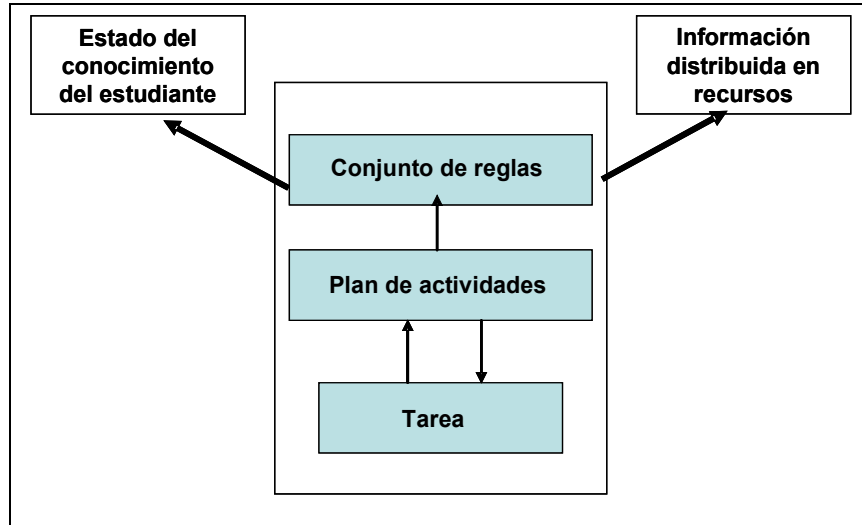


Figura 2.4 Arquitectura básica de un agente de APA-Prolog.

Adis usa información explícita inicial que solicita al estudiante a través de formularios y la almacena en una base de datos relacional. Posteriormente, la información puede actualizarse a través de dos vías: la primera, a partir de la interacción entre el sistema y el usuario y la segunda, de la misma forma que la inicial.

Los datos que maneja Adis se concibieron abiertos y dinámicos. Dinámicos porque pueden cambiar a partir de la actuación del estudiante o a partir de su propio metaconocimiento y abiertos porque no está restringido al uso del profesor o al sistema solamente sino que se expone al usuario para que esté consciente de cómo es visto por el sistema y pueda reflexionar y tomar decisiones que lo animen a mejorar su desempeño.

Para lograr todo lo anterior Adis desarrolla las siguientes actividades:

1. Apropiarse del nombre y la palabra clave que usará el estudiante para autenticarse.
2. Capturar el grado de conocimiento que posee el estudiante sobre cada uno de los conceptos fundamentales del ambiente y recordarlos (véase la figura 2.5).

3. Investigar si el estudiante tiene algún objetivo específico que vencer, por ejemplo: los hechos, las reglas, las preguntas, etc.
4. Almacenar esos objetivos específicos.
5. Facilitar al aprendiz los elementos necesarios para autoevaluarse.

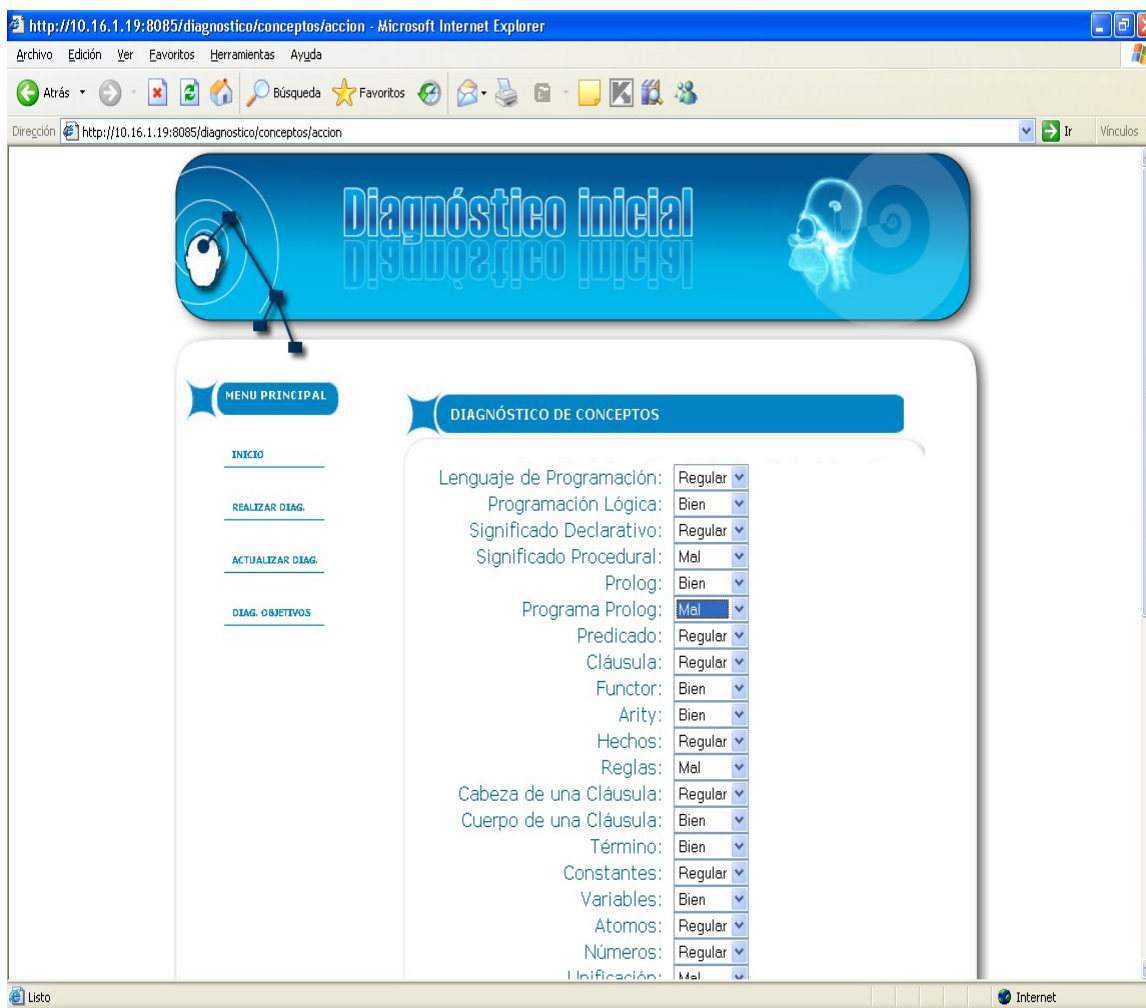


Figura 2.5 Formulario que permite al estudiante autodiagnosticarse.

Esta última actividad merece un comentario adicional, pues generalmente los estudiantes no saben medir sus conocimientos sobre algunos temas y responden bien, regular o mal teniendo en cuenta sólo lo que han leído o han escuchado al respecto. Sin embargo, Adis pretende actuar en el campo del metaconocimiento del estudiante, diciéndole por ejemplo: “... para tener un grado de conocimiento bueno acerca de las reglas usted debe saber, ante un problema dado, escribir las reglas correspondientes; si no sabe hacer esto pero es capaz

de interpretar las ya existentes, entonces su conocimiento es regular y si no puede hacer ninguna de las dos entonces es mal”. La figura 2.6 muestra la arquitectura de Adis, que constituye un caso particular de la arquitectura básica pues este agente no contiene reglas ni se relaciona con la información a mostrar. Para hacer esta información perdurable se implementó una base de datos relacional en MySQL.

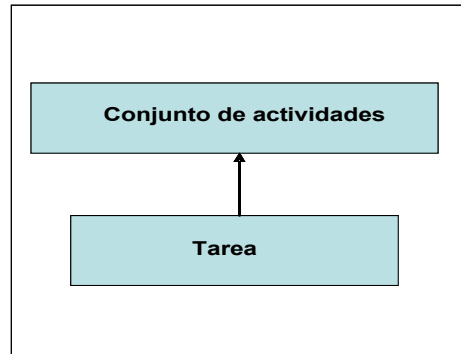


Figura 2.6 Arquitectura de Adis

La base de datos la usan, como fuente de información, los agentes Tivo, Teo, Tica y Eva para conocer el estado de conocimiento del estudiante con que están trabajando. Eva también tiene autoridad para actualizarla cuando el aprendiz haya obtenido resultados que impliquen cambios en su evaluación y pueda, por tanto, tener un estado de conocimiento diferente. La figura 2.7 ilustra esa relación. En el epígrafe 2.5 se expone cómo se logra este intercambio a través del modelo del pizarrón.

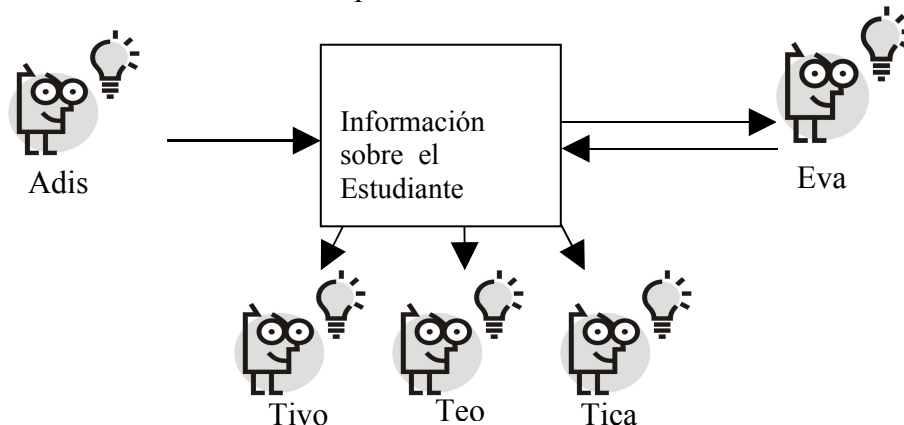


Figura 2.7 Interacción de los A.I. con la información acerca del estudiante.

Se decidió usar MySQL porque es uno de los gestores de base de datos de código abierto más populares disponibles actualmente. Es seguro, rápido y fácil de usar. Fue desarrollado principalmente para manejar grandes bases de datos mucho más rápido que las soluciones existentes. Su conectividad, velocidad y seguridad lo hacen un servidor especialmente apropiado para bases de datos y aplicaciones en Internet, la aplicación que se construye actúa a distancia y por tanto se precisa que la demora no sea significativa. MySQL puede integrarse tanto al ambiente Windows como Linux, permitiendo la migración y creación de aplicaciones para ambos sistemas operativos. Su instalación es muy fácil y bien documentada. Es compatible con múltiples plataformas, incluyendo todas las versiones de Windows. Consume muy pocos recursos, tanto de CPU como de memoria. Posee en su distribución herramientas para chequear sus procesos, sus consultas y para recuperar datos corruptos, entre otras (Díaz, 2006).

2.2.2 Agente Tivo

El agente Tivo tiene por objetivo facilitar el camino hacia informaciones específicas basadas en insuficiencias puntuales, para las cuales recomienda los caminos y recursos a utilizar, por ejemplo, si se conoce que un estudiante tiene dificultades en el concepto de lista, Tivo podría dar la siguiente recomendación: "... debes revisar el mapa llamado Listas, consultar el simulador de Listas y los materiales M1, M2 y M3...".

El agente no ayuda al estudiante a construir su conocimiento, sólo le facilita los materiales necesarios para ello, acortándole el tiempo de navegación y evitando que se pierda y deje de utilizar algunos recursos que pudieran aportar conocimientos significativos.

Tivo proporciona una navegación adaptativa para lo cual desempeña tres actividades fundamentales:

1. Propone una lista, ordenada por relevancia, de todos los materiales disponibles en APA-Prolog que pudiera ayudar a las necesidades cognitivas del estudiante.
2. Muestra la secuencia de pasos para lograr el objetivo actual, sugiriendo los enlaces más importantes a seguir, por ejemplo, si el objetivo son los predicados, el camino indicado puede ser: Prolog → Programas → Predicado.

3. Alerta acerca de los conceptos que son precedentes al concepto objetivo, por ejemplo, si el objetivo fuera predicados, los prerrequisitos serían los conocimientos relacionados con las cláusulas, el functor y el arity de las cláusulas, entre otros.

2.2.3 Agente Teo

Los elementos teóricos son de suma importancia en cualquier proceso de enseñanza-aprendizaje. El agente Teo es capaz de decidir qué materiales teóricos, de todos los que hay disponibles en el ambiente, son los adecuados para un estudiante en particular. Para ello se apoya en la representación del conocimiento del estudiante y realiza las siguientes acciones:

1. Usando las técnicas de mapas adaptativos, y ocultamiento adaptativo de enlaces determina qué variante de mapas mostrar, adicionando o quitando enlaces para mostrar una configuración adaptada a su nivel.
2. Decide qué textos presentar en cada concepto, implementando para eso las siguientes técnicas:
 - a) Variantes de páginas: El sistema almacena dos o tres versiones de una misma página, estas se diferencian en la cantidad de información que contienen y en la profundidad con que es tratada.
 - b) Explicación de variantes: El contenido tiene diferentes versiones de explicaciones que se muestran de acuerdo al estado de conocimiento del estudiante.
 - c) Coloca en orden descendente de relevancia los enlaces a la información más adecuada según los conocimientos previos del estudiante.
3. Determina los ejemplos a visualizar, valiéndose de las técnicas anteriores.
4. Suministra una lista ordenada de bibliografía recomendada.
5. Sugiere el orden del próximo concepto a visitar desde el nodo actual.
6. Inserta definiciones de todos los conceptos presentes entre los prerrequisitos para cada nodo.
7. Recomienda ejercicios para entrenamiento y en ocasiones el uso de entrenadores.

8. Recomienda ver simuladores.
9. Recomienda evaluaciones.
10. Motiva para el próximo concepto a visitar.

Para desarrollar con éxito la tarea asignada, Teo manipula diferentes recursos, basándose en un conjunto de reglas que tienen el formato general siguiente:

```
Si estado de conocimiento (C) = <evaluación>
    entonces mostrar <medio>
```

donde:

C representa el concepto que se está tratando,

<evaluación> toma valores del conjunto {bien, regular, mal},

<medio> representa un subconjunto (diferente del vacío) de los recursos disponibles para Teo (mapas conceptuales, textos explicativos, simulaciones, presentaciones, tutoriales)

2.2.3.1 Los Mapas Conceptuales

Los mapas conceptuales se manejan como un recurso más dentro de APA-Prolog. La primera versión del sistema estaba constituida por un solo mapa que contenía todos los recursos, pero la práctica y el desarrollo propio de la investigación demostró que esta solución no era adecuada debido a que la gran cantidad de información y recursos que contenía abrumaba a los estudiantes. Por ese motivo, el mapa inicial se fragmentó en un conjunto de submapas que son más pequeños y responden a contenidos específicos del objeto de estudio. Las figuras 2.8 y 2.9 muestran dos de esos mapas y también se puede apreciar que cada concepto puede tener asociado uno o más iconos que son enlaces a los diferentes recursos del sistema.

Existen diversas topologías para representar los mapas (araña, jerárquica, de flujo y otras). Los mapas de APA-Prolog se organizan de manera jerárquica, la elección de la topología se basa en el hecho de que se pueden hacer análisis en dos sentidos: de lo general a lo particular y de lo particular a lo general. Por otra parte, facilitan la inclusión de nuevos conceptos con lo que se puede lograr sesiones de trabajo colaborativo que ayudan a que los estudiantes compartan conocimientos y confronten sus ideas.

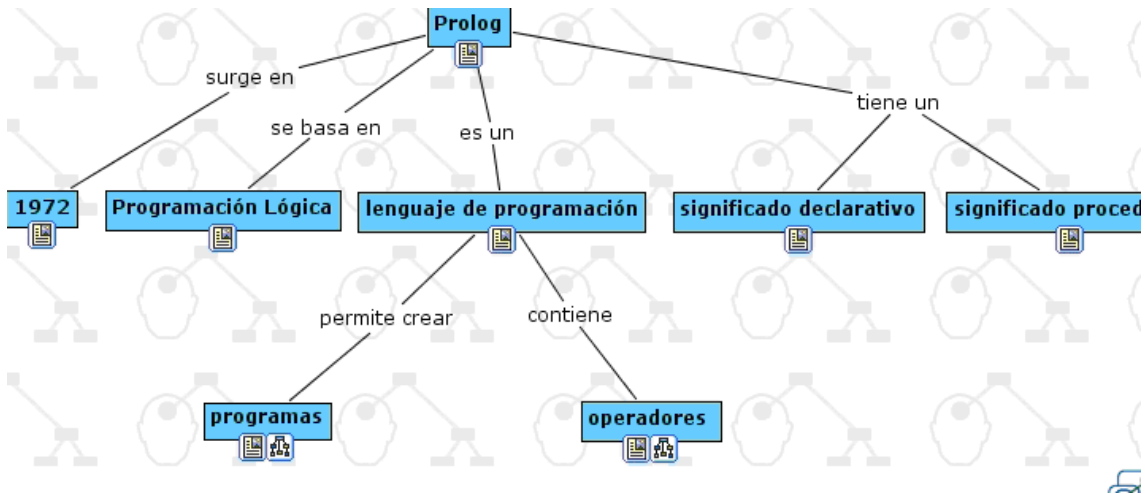


Figura 2.8 Mapa Conceptual que brinda las nociones más generales sobre Prolog.

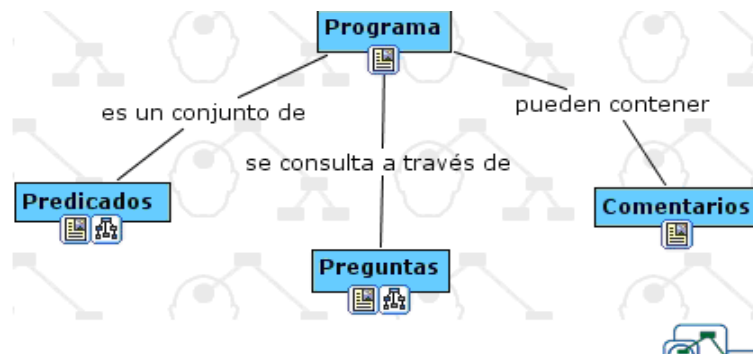


Figura 2.9 Mapa conceptual derivado del nodo Programas de la Figura 2.7.

Para desarrollar la aplicación pueden tomarse dos variantes; una es desarrollar una herramienta propia que ayude a elaborar los mapas conceptuales, la autora considera que esa opción no se justifica por el hecho de existir una gran cantidad de recursos informáticos con ese fin y la otra es tomar una de esas herramientas y adaptarlas a las necesidades exigidas en el proyecto.

Las herramientas de desarrollo o editores de mapas conceptuales se pueden clasificar en dos categorías generales:

- Gratuitas y sujetas a licencias de pocas exigencias.
- De uso privado, por las que hay que pagar un precio determinado y someterse a una licencia comercial de uso.

Tomando en cuenta las particularidades del Sistema de Enseñanza Cubano, la herramienta escogida tiene que pertenecer a la primera categoría y debe tener, además, una licencia que no comprometa la continuidad del proyecto.

Después de evaluar varios software de desarrollo de mapas conceptuales se escogió al CmapTools¹⁹ como la mejor opción entre las ofertas gratuitas. Este software, desarrollado por el IHMC²⁰ se diseñó con el objetivo de apoyar la construcción de modelos de conocimiento representados en forma de “Mapas Conceptuales” aunque también pueden elaborarse con él “Telarañas”, “Mapas de Ideas” y “Diagramas Causa-Efecto”, todos dentro de un entorno de trabajo intuitivo, amigable y fácil de utilizar. CmapTools reúne todas las características que debe tener un software para cumplir con los objetivos de aprendizaje a través de las Tecnologías de la Información y las Comunicaciones.

Una vez organizada la información se hace necesario profundizar en ella. En APA-Prolog los textos explicativos y los tutoriales se diseñaron con este objetivo.

2.2.3.2 Textos explicativos

La mayoría de los conceptos de la aplicación desarrollada tienen asociados, al menos, un texto que puede contener:

- a) Explicaciones de ¿en qué consiste?, ¿para qué se utiliza?, ¿cuál es su sintaxis?, ¿qué elementos lo conforman?, etc.
- b) Ejercicios resueltos y detalladamente explicados. En este caso están todos los que se asocian al concepto Ejercicios Resueltos, entre los que se pueden citar, problemas clásicos de IA como el de las ocho reinas y las torres de Hanoi con variantes de solución.

¹⁹ CmapTools. Herramientas cliente servidor para el desarrollo de mapas conceptuales.

<http://cmap.ihmc.us/conceptmap.html>

²⁰ Institute for Human and Machine Cognition. <http://cmap.ihmc.us/conceptmap.html>

- c) Direcciones de sitios importantes relacionados con la Programación Lógica y el Prolog.
- d) Datos biográficos de importantes personalidades asociadas al surgimiento y desarrollo del Prolog.
- e) Datos históricos, como los asociados al CUSS, al surgimiento del Prolog, y otros.
- f) Definiciones, por ejemplo, de los diferentes estilos de programación, permitiendo a los estudiantes comparar y valorar las particularidades del paradigma estudiado.
- g) Bibliografía propuesta.

Muchos de estos textos cuentan con tres versiones, atendiendo a los diferentes niveles de detalles que el aprendiz precisa (básico, medio, detallado). Todos se encuentran en formato Web estándar, por lo tanto se puede acceder a ellos fácilmente y no requieren de un navegador específico.

2.2.3.3 Simuladores

La Real Academia define simulador como: “aparato que reproduce el comportamiento de un sistema en determinadas condiciones, aplicado generalmente para el entrenamiento de quienes deben manejar dicho sistema”.

El uso de los simuladores computarizados data de la segunda mitad del siglo pasado. Se comparte el criterio de (Martínez, 2007) al considerar que actualmente el uso de los simuladores está mayormente dado por la posibilidad de repetir, en condiciones idénticas y a partir de su modelación, procesos y fenómenos algo difíciles de lograr u observar en condiciones reales y, por tanto, estudiar sistemáticamente sus comportamientos hasta lograr los objetivos deseados. Por otro lado, los simuladores eliminan riesgos que casi siempre se presentan en la interacción con la realidad. Todo esto ha hecho posible su utilización creciente en distintas ramas del quehacer humano, entre las que se distinguen: la aviación, los programas espaciales, la medicina y la educación.

Los simuladores educativos se incluyen dentro de un conjunto de software de apoyo curricular, generalmente denominado software educativo que busca reforzar, complementar o servir de material docente en diversas asignaturas.

El ambiente construido pone a disposición de los usuarios tres simuladores asociados a importantes mecanismos del lenguaje Prolog que son difíciles de estudiar y comprender: el backtracking, el matching y el trabajo con listas.

Los objetivos comunes a los tres simuladores son:

1. Simular la forma en que Prolog trabaja, de modo particular cómo hace el backtracking, cómo se lleva a cabo el proceso de matching y cómo se efectúan las operaciones fundamentales con listas.
2. Ayudar a entender tales conceptos.
3. Disponer de un conjunto de programas ilustrativos cuya trayectoria de ejecución se muestra y puede ser seguida por los estudiantes.
4. Contribuir a desarrollar un adecuado estilo de programación lógica.

Según (Lezcano y Valdés, 1998) pudiera pensarse que una herramienta que presenta el funcionamiento interno de un lenguaje declarativo, no es didácticamente la más adecuada, pero ellos mismos concluyen en sus investigaciones el haber comprobado que una vez que los estudiantes comprenden la potencia de los mecanismos del lenguaje, aprovechan mejor y a plenitud sus posibilidades. Esta afirmación ha podido ser confirmada en la presente investigación de acuerdo a las experiencias propias desarrolladas en el aula con el uso de los simuladores.

Los requerimientos funcionales que se tuvieron en cuenta durante el proceso de diseño e implementación de los simuladores y que fomentan el aprendizaje activo y por descubrimiento, básicamente fueron:

1. Permitir al usuario escoger el ejemplo que desee.
2. Mostrar el código del ejemplo seleccionado para que los estudiantes vean como está implementado con el propósito de mejorar los hábitos de programación.

3. Mostrar las variables involucradas y los valores que estas van tomando.
4. Dejar un historial del código accedido en cada paso.
5. Mostrar una explicación detallada del ejemplo.
6. Ejecutar paso a paso y al ritmo del usuario el ejemplo seleccionado.
7. Permitir regresar a pasos anteriores para aclarar dudas pendientes.

Basado en el análisis anterior, se obtuvieron tres aplicaciones con una interfaz homogénea, amigable y sencilla donde todas las opciones son accesibles desde la ventana principal. La figura 2.10 muestra la apariencia del simulador de backtracking (muy similar a la de los otros dos simuladores). Se pueden apreciar: un menú, cuyas opciones son cuatro ejemplos ordenados ascendentemente de acuerdo a su complejidad, una pequeña barra de herramientas con los botones para comenzar y terminar, a continuación la ventana se divide en dos paneles, el izquierdo tiene en la parte superior la barra de consulta donde aparece explícitamente la consulta que se desea hacer al programa seleccionado, seguidamente aparecen los hechos y reglas correspondientes al ejemplo y un cursor en forma de mano que va indicando por dónde va la ejecución del programa, mientras en el panel derecho se observa una explicación detallada de cada paso dado por Prolog para obtener la solución.

En la parte inferior del panel izquierdo se encuentran las variables involucradas en el proceso y los valores que van tomando, de forma tal que el estudiante pueda seguir activamente todo el algoritmo de solución. Al mismo tiempo estos valores se van almacenando, lo que facilita el análisis y la reflexión en cualquier momento a partir del historial creado.

Es importante destacar dos características de estos simuladores:

1. Son applets que se descargan de forma ligera y solo necesitan para su ejecución que la máquina cliente tenga instalada la máquina virtual de Java.
2. Corren en cualquier tipo de computadora, navegador y sistema operativo.

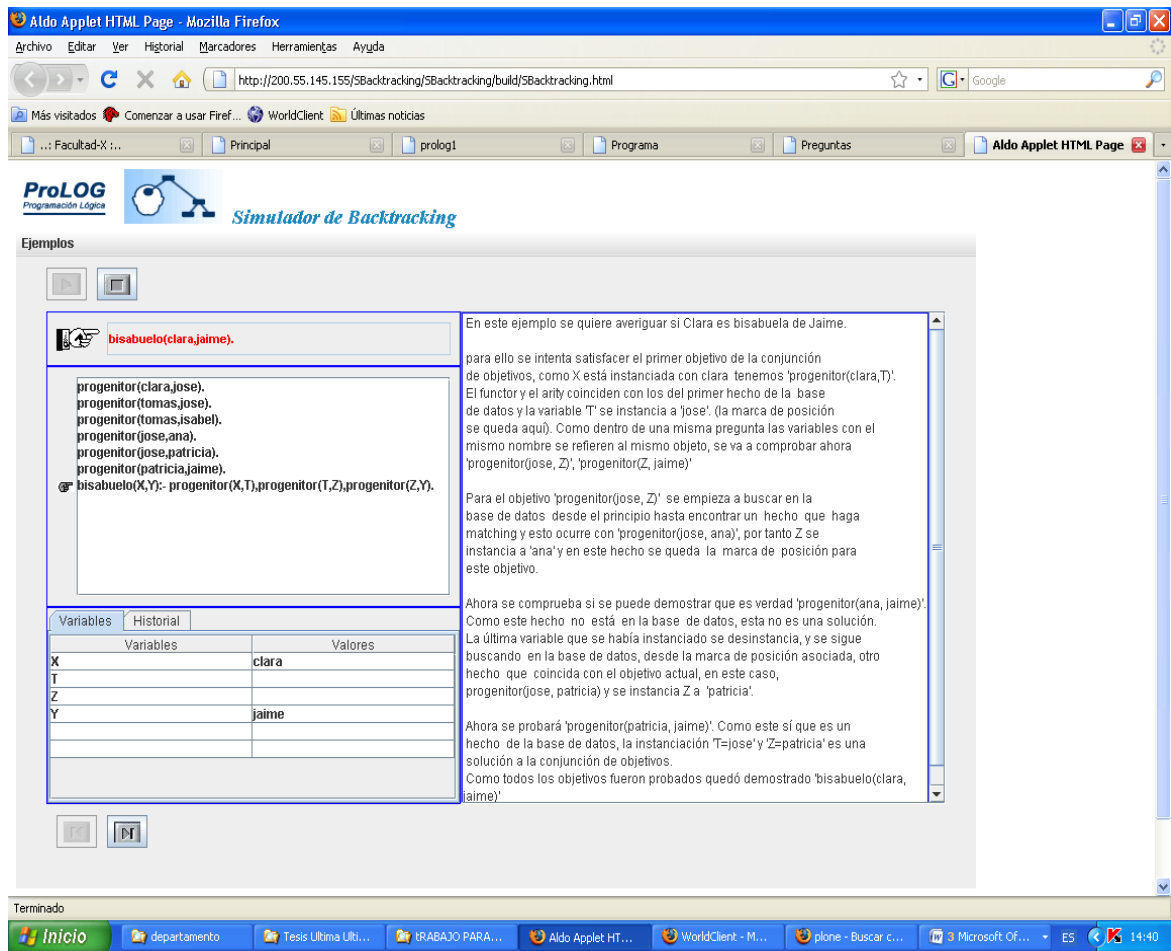


Figura 2.10 Instantánea del simulador de backtracking.

2.2.4 Agente Tica

Tica es el agente encargado de presentar las actividades prácticas que debe realizar el estudiante, las cuales están asociadas a su nivel de conocimiento. El agente realiza las siguientes actividades:

1. Presenta los ejercicios a resolver y las guías para su solución.
2. Ofrece enlaces a prerrequisitos.
3. Refuerza las respuestas correctas.
4. No penaliza las respuestas incorrectas sino que proporciona elementos importantes a tener en cuenta para llegar a la solución correcta (explicaciones adicionales,

explicaciones de prerequisites, explicaciones de variantes y explicaciones comparativas).

5. Si el estudiante no llega a la solución correcta, entonces recomienda estudiar la teoría necesaria para conseguirlo (explicaciones adicionales y variante de páginas).
6. Si el estudiante responde acertadamente el cuestionario se le recomiendan documentos y materiales para profundizar.
7. Recomienda pasar al evaluador.

Los entrenadores son los recursos con que cuenta Tica para realizar sus actividades, ellos están asociados a aquellos conceptos que requieren ser ejercitados.

2.2.4.1 Entrenadores

En ningún ambiente para el aprendizaje, real o virtual deben faltar las actividades prácticas. En esta ocasión se concibieron mediante entrenadores inteligentes esencialmente orientados al desarrollo de habilidades, pues no pretenden la dirección total del proceso de instrucción ni llevan a cabo la formación de conceptos nuevos. Supervisan la actividad práctica del estudiante mediante el control de los errores durante la solución de los ejercicios, hacen recomendaciones y controlan la presentación dosificada de problemas y ejercicios (Medina, 2007).

Teniendo en cuenta la cantidad de entrenadores que pudiera ser necesaria para ambientes de enseñanza-aprendizaje relacionados con la programación y dado el carácter eminentemente práctico que la identifica, se creó una herramienta que puede usarse para desarrollar tantos ejercitadores como se requiera y sin inconvenientes para extrapolarlos a otras asignaturas u otros entornos de aprendizaje. De modo que puede hablarse de dos aplicaciones:

- Una herramienta de Administración (HerAdE), con la que trabaja el profesor o persona encargada de situar los ejercicios debidamente graduados por complejidad.
- El entrenador, con el que interactúan los estudiantes.

HerAdE tiene un sistema de seguridad que permite mantener el control del personal con acceso a:

- Definir los conceptos con que se va a trabajar.
- Añadir problemas.
- Adicionar incisos a los problemas.
- Agregar respuestas correctas e incorrectas a los incisos del problema.
- Establecer el índice con que se comparará el por ciento de respuestas correctas.
- Modificar o eliminar cualquiera de los elementos anteriores.

Esta información se almacena en una base de datos relacional cuyas tablas fundamentales son:

Concepto (identificador_c, descripción)

Problema (identificador_p, nivel_de_dificultad, concepto_al_que_está_asociado)

Incisos (identificador_i, identificador_p, texto_descriptivo)

Respuestas_incisos (identificador_r, identificador_p, identificador_i, texto, categoría)

Donde:

- categoría puede ser: correcta o incorrecta y
- nivel de dificultad puede tomar alguno de los valores: alto, medio o bajo.

Esta herramienta cuenta con dos niveles de administración:

1. El administrador del Zope que tiene facultad para crear nuevos usuarios con privilegios de administración sobre la herramienta antes descrita.
2. Los otros administradores (publicadores) que son los encargados de entrar toda la información relacionada con los ejercicios.

La herramienta para la ejercitación tiene como objetivo facilitar la realización de actividades prácticas por parte de los estudiantes y servirle de guía durante la sesión de trabajo. Hace un balance de la cantidad de incisos resueltos de modo correcto e incorrecto y ofrece una valoración general de la actuación del estudiante frente al contenido tratado. Del mismo modo procede con la cantidad de problemas que el alumno resuelve correcta e

incorrectamente para hacerle las sugerencias y recomendaciones oportunas. Si de modo general las respuestas del estudiante son correctas se le exhorta a pasar al Evaluador. (Véase la figura 2.11)



Figura 2.11 Vista del entrenador para el concepto Reglas.

2.2.5 Agente Eva

El agente Eva tiene los privilegios de evaluar al estudiante y decidir si modifica o no su estado de conocimiento, en caso afirmativo lo modifica, situándole valores de bien, regular o mal. Se desempeña de la siguiente manera:

1. Selecciona el cuestionario teniendo en cuenta el concepto y el estado de conocimiento del estudiante en ese concepto.
2. Evalúa cualitativamente.

3. Recomienda pasar o no al siguiente concepto en el mapa.
4. Sugiere otros recursos para apropiarse del contenido cuando el estudiante no es capaz de responder correctamente alguna pregunta del examen.
5. Orienta bibliografía para profundizar si el estudiante ha respondido correctamente.
6. No penaliza al estudiante si los resultados no son los esperados sino que hace recomendaciones para superar las dificultades encontradas.
7. Si el estudiante responde con éxito el formulario que le corresponde, pasa su estado de conocimiento a un nivel superior (bien o regular).
8. Si el estudiante no responde con éxito el formulario que le corresponde entonces pasa su estado de conocimiento a un nivel inferior (regular o mal).

2.2.5.1 Evaluadores

En cualquier proceso de enseñanza-aprendizaje se distinguen tres grandes momentos: la introducción del nuevo contenido, la ejercitación y la evaluación. Por tanto, no resulta difícil entender que si se asumen transformaciones en el diseño e implementación de los medios de enseñanza, en las formas de organización, en el sistema de tareas y el trabajo independiente de los estudiantes, también hay que llevar a cabo transformaciones en el sistema de evaluación.

Los evaluadores son los medios con que cuenta Eva para llevar a cabo su trabajo. Un evaluador es un recurso que tiene la finalidad de brindar un cuestionario, a modo de conclusión, de un concepto específico.

Si dicho cuestionario se llena de forma exitosa, significará que el aprendiz ha pasado a un grado de conocimiento superior lo que se refleja automáticamente en el estado de conocimiento del estudiante sobre ese concepto. Los evaluadores se confeccionaron de modo muy similar a los entrenadores, utilizando la misma aplicación.

Con el propósito de propiciar un ambiente de trabajo cómodo, APA-Prolog dispone de otros dos recursos denominados espacio para el intercambio y directorio temático. A continuación se describen los recursos mencionados.

2.3 Espacio para el intercambio

Según (Rodríguez, Estrada, J. y Castillo, 2008), en cuanto a las estrategias de enseñanza-aprendizaje en entornos virtuales, se reportan como más eficaces las asociadas al método de trabajo colaborativo.

En estos entornos, el aprendizaje colaborativo se entiende como el proceso formativo facilitado por la interacción social en un entorno de comunicación, evaluación y cooperación entre iguales.

Al contar con un entorno virtual de enseñanza-aprendizaje, es posible recurrir a diferentes modelos de comunicación en función de los recursos pedagógicos que se decidan utilizar, las actividades que se propongan en el curso, o el tipo de materiales por los que se opte. Las interacciones de carácter interpersonal que pueden tener lugar en esta clase de entornos de comunicación, según (Rodríguez, Estrada, J. y Castillo, 2008) son:

- Comunicaciones uno-a-uno.
- Comunicaciones uno-a-muchos,
- Comunicaciones muchos-a muchos.

El espacio para el intercambio contribuye a eliminar la sensación de aislamiento que es una de las características más deseables de los procesos de formación a distancia, como también lo es, hacer posible un aprendizaje colaborativo.

El espacio para el intercambio que brinda APA-Prolog tiene cuatro opciones fundamentales: <Conversación en línea>, <Espacio para el debate>, <Noticias> y <Eventos>.

La opción conversación en línea, conocida comúnmente como chat, permite interconectar estudiantes de diferentes lugares del país y del mundo en torno a temas de conversación de diversa índole sobre la Programación Lógica. (Figura 2.12)

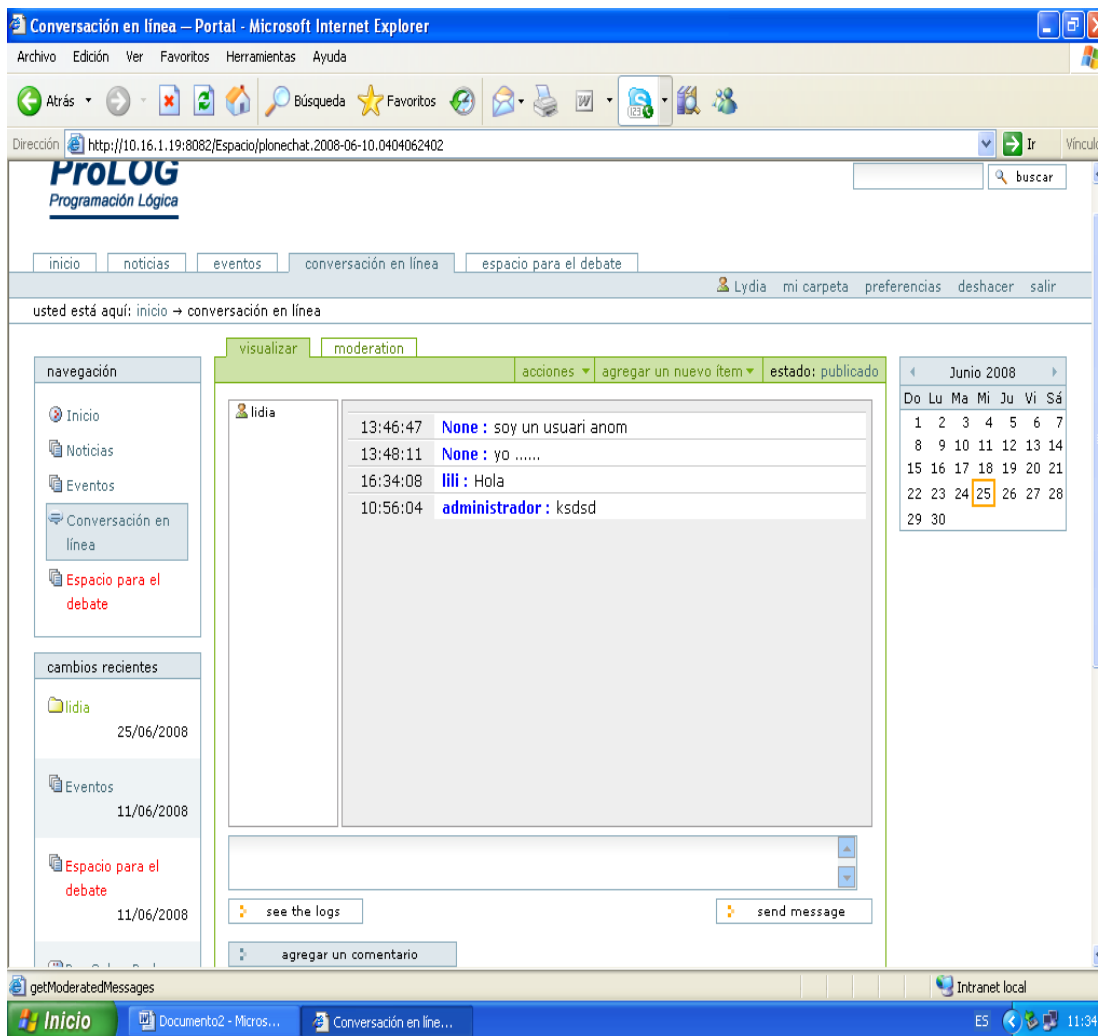


Figura 2.12 Apariencia de la opción Conversación en Línea.

El espacio para el debate se corresponde con los llamados foros de discusión, esta es una de las actividades colaborativas centrales en casi todas las plataformas interactivas pues permite la reflexión colectiva sobre determinados temas, ejercicios, documentos, etc.

El profesor puede establecer el foro para promover la interacción entre los estudiantes. Los materiales, las opiniones, las reflexiones de los aprendices pueden ser enviados a otros participantes y estos pueden enviar sus comentarios o preguntas a todo el grupo y cada uno

de los otros puede responder libremente. En este caso, el espacio prevé la asistencia de un moderador con el fin de conducir correctamente el desarrollo del debate.

La opción Noticias permite que el profesor o administrador publique las noticias que considere necesarias relacionadas con los avances de la Programación Lógica, nuevas versiones de Prolog, nuevas aplicaciones, etc., así como aquellas vinculadas con el desarrollo de las actividades formativas.

Este espacio para el intercambio también permite agrupar y publicar la información relacionada con eventos relevantes para el grupo como pueden ser reuniones, actividades presenciales, actividades de carácter científico y otras. Del mismo modo que las noticias, el administrador las publica y los estudiantes pueden emitir criterios mediante comentarios.

Cada una de las opciones descritas supone la existencia de un líder en el grupo, es decir, un profesor, administrador o facilitador con sus correspondientes privilegios y de una comunidad de educandos.

Este espacio para el intercambio fue construido usando el CMS Plone²¹, sobre un servidor Zope²².

2.4 Directorio Temático

Sin un aseguramiento bibliográfico adecuado no es posible un proceso de enseñanza-aprendizaje exitoso. A veces, por problemas de disponibilidad, de accesibilidad y otros, resulta complejo obtener un libro que se necesita o aún el material preciso que se busca.

Lo anterior se ratifica por encuestas realizadas a los estudiantes, de forma que el asunto se convierte en una prioridad para los docentes y en esa tarea surge la necesidad de emplear herramientas que permitan organizar la información ya localizada, seleccionada y procesada con el fin de hacerla llegar a los estudiantes sin limitaciones de horario ni espacio.

²¹ <http://plone.org/>

²² <http://www.zope.org/>

Un directorio temático especializado es un sistema de información que selecciona, describe y clasifica recursos Web aplicando criterios propios de la bibliotecología y documentación dirigidos a la comunidad académica y científica, basados en la selección de recursos de información de alta calidad, identificados y evaluados por profesionales de la información y especialistas del tema, estos recursos pueden presentarse dentro de una página Web propia, como servicio de un portal bibliotecario o formando parte de los contenidos de un portal científico (Rodríguez, 2007).

Los objetivos de este recurso (Figura 2.13) en el presente trabajo son:

1. Propiciar un lugar al que los estudiantes puedan acudir para revisar publicaciones de diversos tipos que han sido previamente revisadas y aprobadas por un colectivo de profesores que certifican su calidad. Además suple la carencia de Internet en muchos lugares del país y se propicia un uso mas adecuado del ancho de banda al disponer de un repositorio local.
2. Ofrecer mecanismos para introducir y editar, fácilmente la información.

En el directorio temático de APA-Prolog existen informaciones en soportes digitales tradicionales y en forma de hipertexto. La aplicación brinda las funcionalidades básicas del servicio de búsqueda y almacenamiento de recursos y permite:

- La identificación del alumno a través de un sistema de autenticación personalizado, lo que hace posible el seguimiento de las trazas de búsquedas y acceso a la información de los estudiantes.
- El empleo, por parte de los profesores, de los materiales de orientación del estudio independiente.

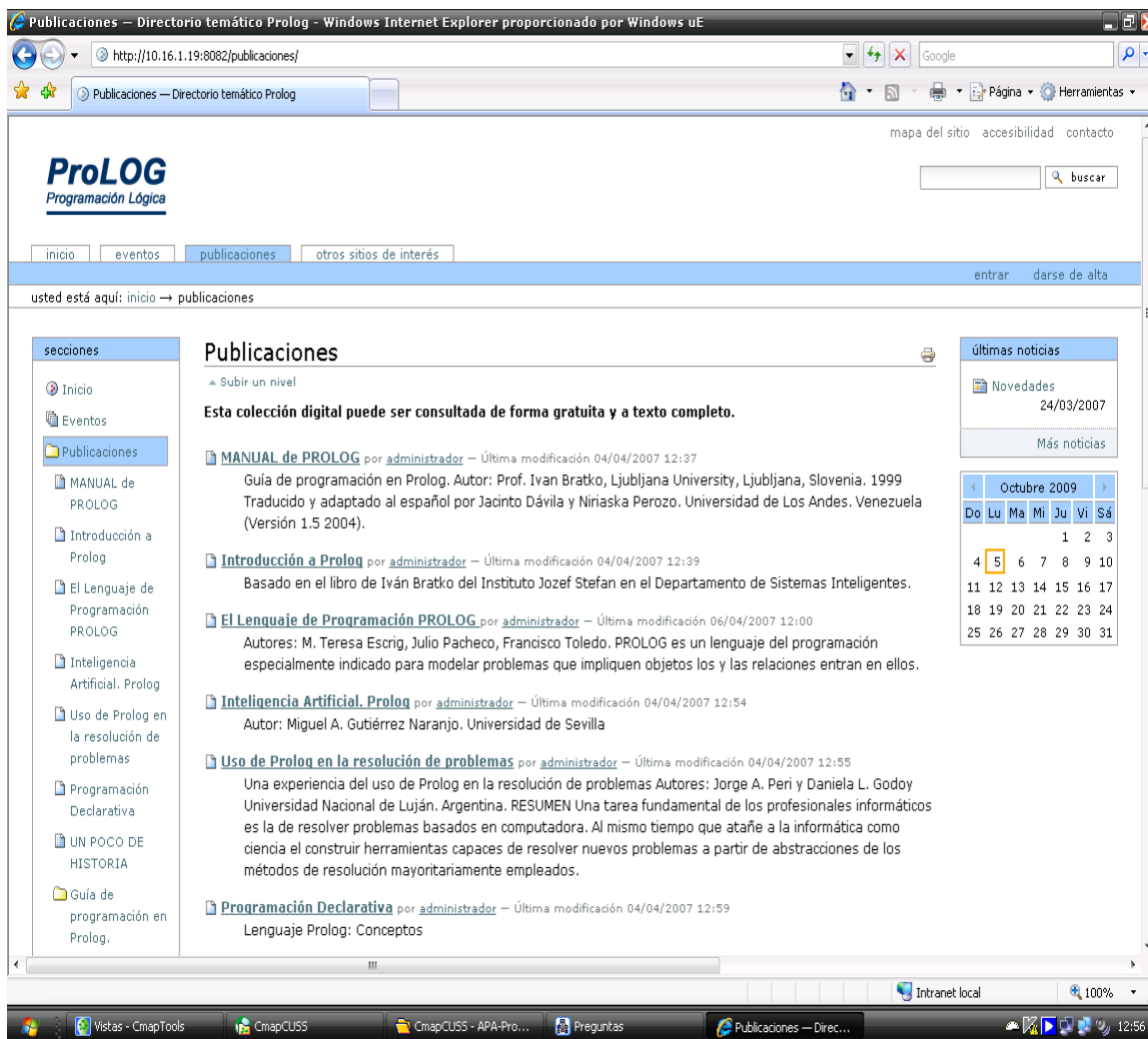


Figura 2.13 Opción Publicaciones del Directorio Temático de APA-Prolog.

- Facilidades para visualizar, descargar o imprimir materiales.
- La atención a diferencias individuales, situando escritos comprensibles para estudiantes con diferentes niveles de asimilación.
- La incorporación de ponencias u otros trabajos desarrollados por lo propios estudiantes lo cual les sirve de estímulo y motivación.

Se identificaron cuatro tipos de usuarios para este recurso:

1. Anónimo: Es el usuario que no se ha autenticado en el sitio y solo puede:
 - a) Consultar información.

- b) Sugerir información.
 - c) Realizar búsquedas.
2. Alumno: Es el usuario que ha sido autenticado en el sitio y tiene ciertos privilegios como:
- a) Crear una carpeta personal y guardar información para próximas visitas.
 - b) Modificar sus datos como usuario.
 - c) Ver lista de usuarios restantes.
3. Profesor: Es el encargado de:
- a) Confirmar el registro de un alumno como miembro del grupo de trabajo.
 - b) Proponer bibliografía o información a publicar al administrador.
 - c) Revisar las carpetas de sus alumnos.
 - d) Modificar los datos de los alumnos.
4. Administrador: Goza de privilegios globales. Tiene el control absoluto del sitio.

El directorio también brinda otros servicios como Eventos, Efemérides, Novedades, Enlaces a otros sitios de interés y una encuesta que recoge opiniones sobre la calidad del sitio y que permite su constante mantenimiento.

2.5 Modelo para la solución cooperativa

Como ya se ha expuesto, APA-Prolog basa su trabajo en la actuación de cinco Agentes Inteligentes con capacidad para ejecutar diversas técnicas de adaptación. Estos agentes tienen vida propia pero interactúan entre sí para ofrecer contenidos personalizados e influir en la motivación del aprendizaje mediante el ofrecimiento de contenidos de retroalimentación, ejercicios adaptados al nivel de conocimiento y una interfaz atractiva.

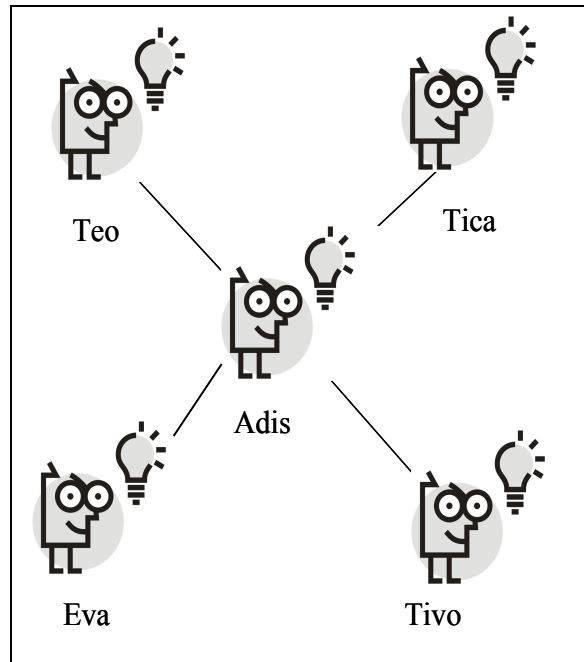


Figura 2.14 Interacción entre los agentes.

El agente Adis obtiene la información primaria acerca del estado de conocimiento del estudiante, esa información la necesitan los demás agentes para realizar su trabajo, de ahí la necesidad de la existencia de relaciones de comunicación entre Adis y los restantes agentes (Figura 2.14). La vía para interactuar con la información se apoya en el modelo del pizarrón (Figura 2.15).

El pizarrón, para el conjunto de agentes que sustentan el trabajo de APA-Prolog, está representado por la clase Estudiante, de modo que, cuando un aprendiz intenta entrar al ambiente lo primero que se hace es verificar si ya se diagnosticó (este hecho implica que ya existe como usuario). En ese caso:

1. Se crea una instancia de la clase Estudiante y se llena con todos los estados de conocimiento del estudiante que se obtienen de una consulta a la base de datos.
2. Se crea una sesión con dos objetivos: el primero, poder llevar la información de la instancia creada de una página Web a la otra de forma que en cada página se pueda preguntar por el estado del conocimiento acerca de un concepto determinado; el segundo, que varios usuarios puedan trabajar a la misma vez en sesiones diferentes.
3. Cuando el estudiante abandona el sistema se actualiza la base de datos a partir de la instancia de la clase y se cierra la respectiva sesión.

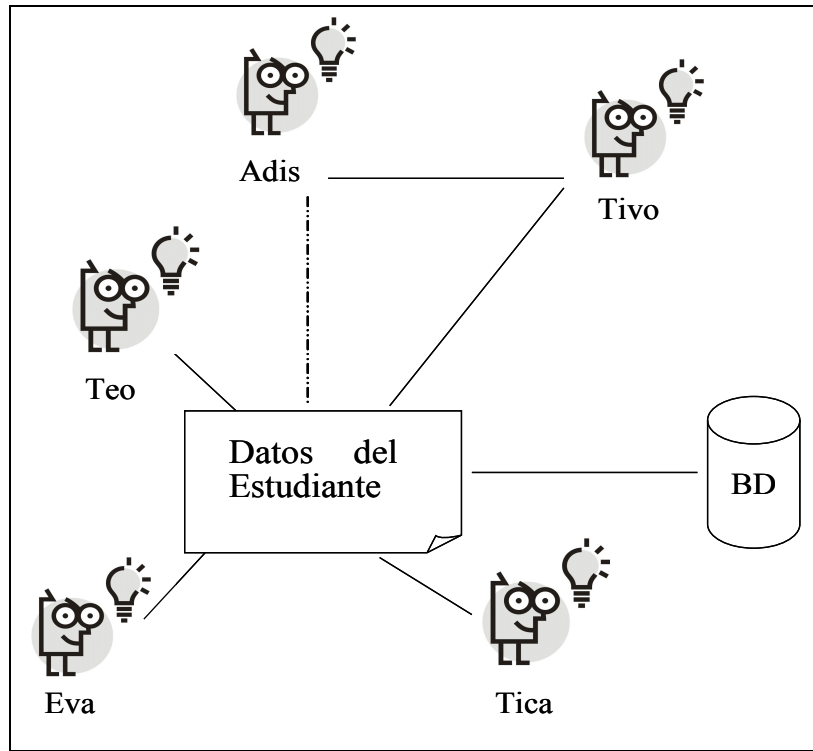


Figura 2.15 Modelo para la solución cooperativa en APA-Prolog.

De forma que la conducta general del sistema multiagente es como se ilustra en la figura 2.16 y el funcionamiento general del ambiente, una vez diagnosticado el estudiante como se muestra en la figura 2.17.

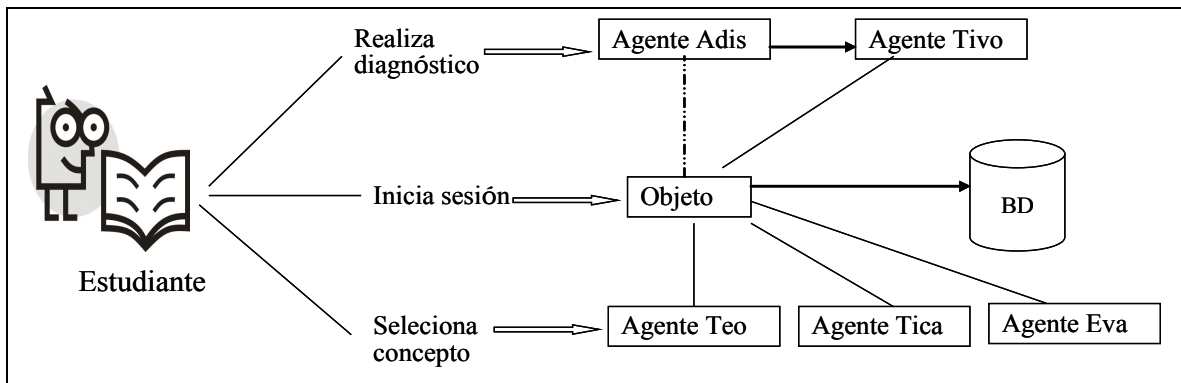


Figura 2.16 Conducta del sistema multiagente en APA-Prolog

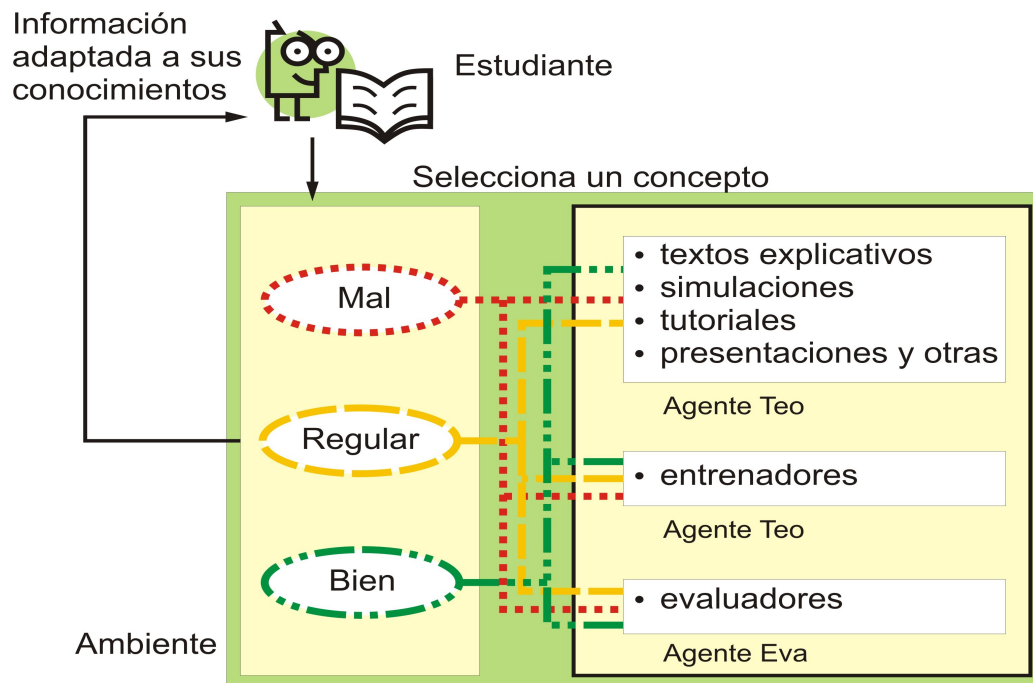


Figura 2.17 Representación del funcionamiento general del sistema.

Este funcionamiento hace posible que, para un estudiante cuyo estado de conocimiento sobre el concepto Programa Prolog sea regular se muestre un mapa con los recursos que se aprecian en la figura 2.18 y para un estudiante evaluado de mal en el mismo concepto se muestre un mapa como el de la figura 2.19.

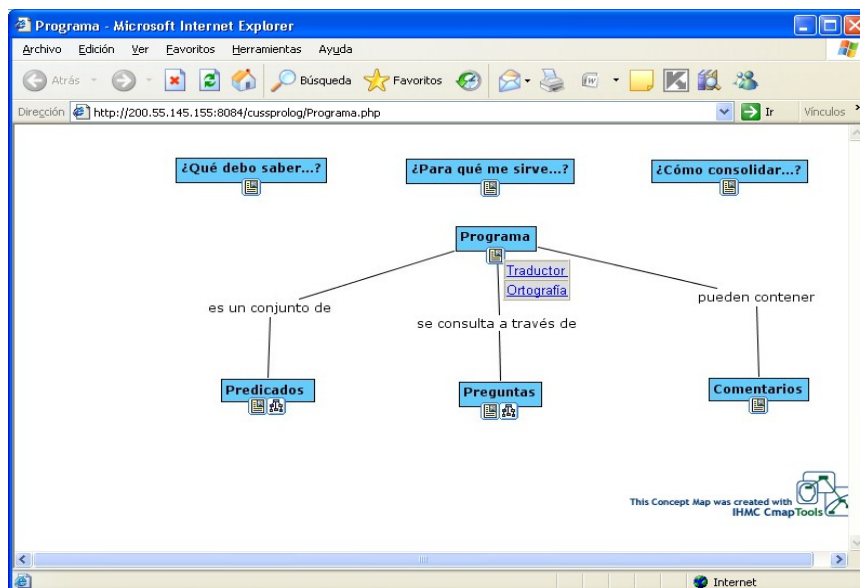


Figura 2.18 Mapa conceptual para un estudiante evaluado de regular.

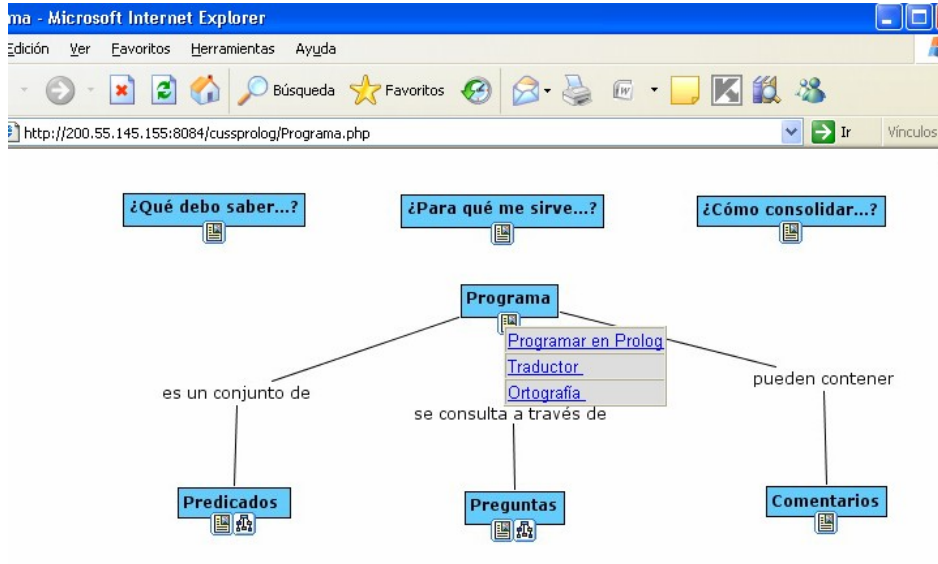


Figura 2.19 Mapa conceptual para un estudiante evaluado de mal.

2.6 Conclusiones Parciales

El capítulo describe al sistema multiagente como eje central de APA-Prolog. Cada uno de sus agentes maneja los recursos que le corresponden y estos llegan al estudiante adaptados a sus necesidades.

Los mapas conceptuales empleados como interfaz confieren a los usuarios de APA-Prolog una representación esquemática de todo el contenido por aprender, haciéndoles conscientes de cuánto han vencido y cuánto les falta por vencer.

En el capítulo siguiente se hará un análisis de los resultados de las encuestas aplicadas a los estudiantes para medir la efectividad de la solución propuesta, así como los resultados de encuestas aplicadas a expertos en el tema para evaluar la calidad del recurso creado.

Capítulo 3

Metodología para construir sistemas
multiagentes inteligentes basados en
Web. Algunas aplicaciones

3. METODOLOGIA PARA CONSTRUIR SISTEMAS MULTIAGENTES INTELIGENTES BASADOS EN WEB. ALGUNAS APLICACIONES

El Ambiente de enseñanza-aprendizaje para la Programación Lógica que se propone puede emplearse de diferentes formas, dependiendo de los objetivos del usuario en particular. Está especialmente concebido para la formación de Ingenieros en Informática pero puede ser usado por estudiantes de Licenciatura en Ciencia de la Computación o cualquier otra persona interesada en aprender este estilo de programación al estar disponible a través de la Intranet del MES o Internet.

En este capítulo se exponen los resultados generales de la aplicación del sistema elaborado y con ese propósito se ha dividido en cinco partes esenciales que se trazan los objetivos siguientes:

- 1) Comunicar la experiencia del empleo de APA-Prolog en el CUSS.
- 2) Exponer una metodología para la creación de ambientes de enseñanza-aprendizaje basados en mapas conceptuales que toma en cuenta los conocimientos previos del estudiante.
- 3) Analizar los resultados de la evaluación de la efectividad del ambiente.
- 4) Analizar los resultados valorados por expertos.

3.1 Experiencia del empleo de APA-Prolog en el CUSS

APA-Prolog fue utilizado por estudiantes y profesores de la asignatura Programación Descriptiva que se imparte en 4^{to} año de la carrera Ingeniería Informática en el Centro Universitario José Martí de Sancti Spíritus, durante el primer semestre del curso 2008-2009. Se empleó fundamentalmente para complementar el proceso de autoaprendizaje de los educandos, siempre orientado y controlado por el docente.

Su uso estuvo marcado por un proceso de retroalimentación y mejora continua que puede ser vista en dos etapas marcadas por la actividad de los estudiantes: en la primera, actuaron de forma pasiva, recibiendo información, mientras en la segunda etapa contribuyeron a la mejora del sistema aportando nuevos recursos informáticos que fueron construidos, entre otras cosas, con la ayuda de los conocimientos adquiridos en su navegación por el sistema. Los resultados de esta experiencia fueron recopilados mediante entrevistas. El proceso de retroalimentación del sistema se ilustra en la figura 3.1

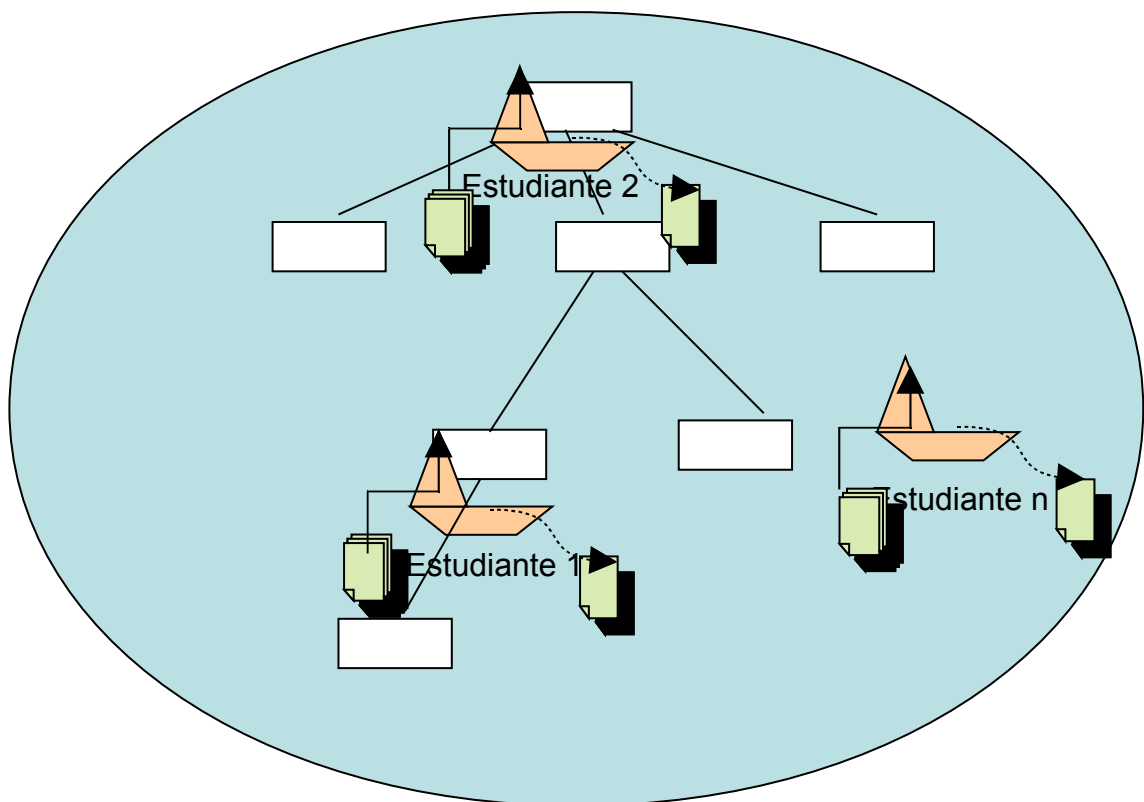


Figura 3.1 Navegación activa de los estudiantes sobre APA-Prolog.

Entre las contribuciones de los estudiantes se destacan las siguientes:

- a) Bases de conocimientos con posibilidades de ser consultadas que contienen información real sobre diversas temáticas de interés común, entre las que se destacan las relacionadas con:
 - La prevención de enfermedades de transmisión sexual.
 - La vida y obra de José Martí.
 - Diferentes especies de mamíferos propias de Cuba.

- Las efemérides más importantes tomadas en cuenta en nuestro país.
 - El tratamiento de algunas enfermedades frecuentes en los Centros de Atención Primaria de la Salud.
 - Algunas enfermedades no muy comunes en nuestro territorio nacional.
- b) Ejercicios resueltos de utilidad práctica, tales como:
- Un traductor inglés-español, español-inglés, que cuenta con términos técnicos empleados en el campo de la informática, además de la posibilidad de dar el significado en español de algunos vocablos que no tienen traducción.
 - Un asesor ortográfico virtual, que ayuda a determinar cuando una palabra se acentúa o no.
 - Un reconocedor de personas con malos hábitos de educación formal.
 - Colecciones de ejercicios relacionados con algunos conceptos en particular.
- c) Tutoriales de diversos temas, entre los que cabe mencionar:
- El tutorial sobre SWI-Prolog, (implementación del lenguaje que se usó para poner a punto todos los ejemplos disponibles en el ambiente). Incluye información detallada para todo tipo de usuario, los archivos necesarios para su instalación y la dirección de su sitio en Internet.
 - Otros tutoriales con las preguntas más frecuentes que se hacen los estudiantes sobre el backtracking, el trabajo con estructuras, con listas, etc., estos se caracterizan por la simplicidad del lenguaje de los propios educandos.
- d) Una tabla que agrupa las características y rasgos distintivos de varias implementaciones de Prolog y en casi la totalidad de los casos (salvo aquellos en que hay que pagar para obtenerlos) los archivos para su instalación. Este es un resultado importante que da la posibilidad a estudiantes y profesores de comparar y luego escoger la mejor implementación de Prolog a utilizar en un momento dado.
- e) Un conjunto de pasos a seguir para lograr implementaciones que combinen Object Pascal con Amzi-Prolog, facilitando de esa forma la convivencia entre ambientes imperativos y declarativos, lo que permite aprovechar las ventajas de ambos.

Esta relación puede ser vista en dos sentidos: Ambiente-Estudiante, Estudiante-Ambiente (Figura 3.2) y tiene la implicación importante del estímulo que constituye para un aprendiz que un trabajo de su autoría sea publicado en Internet para que otros aprendices lo analicen logrando un intercambio de sugerencias y críticas. Todo esto tributa al desarrollo de competencias relacionadas con el trabajo colaborativo y el trabajo en grupos, imprescindibles para el buen desempeño de un Ingeniero Informático.

Pudiera pensarse que la inclusión de los aspectos señalados en los incisos a y b no aportan nada al sistema debido a que la herramienta tiene el objetivo específico de enseñar Prolog, pero en esta investigación se ha tratado de presentar un sistema que, aún cuando tiene un objetivo bien definido, incluya algunas adiciones que tienen el propósito de aumentar el acervo cultural de los estudiantes contribuyendo de esa forma a los objetivos educativos generales.

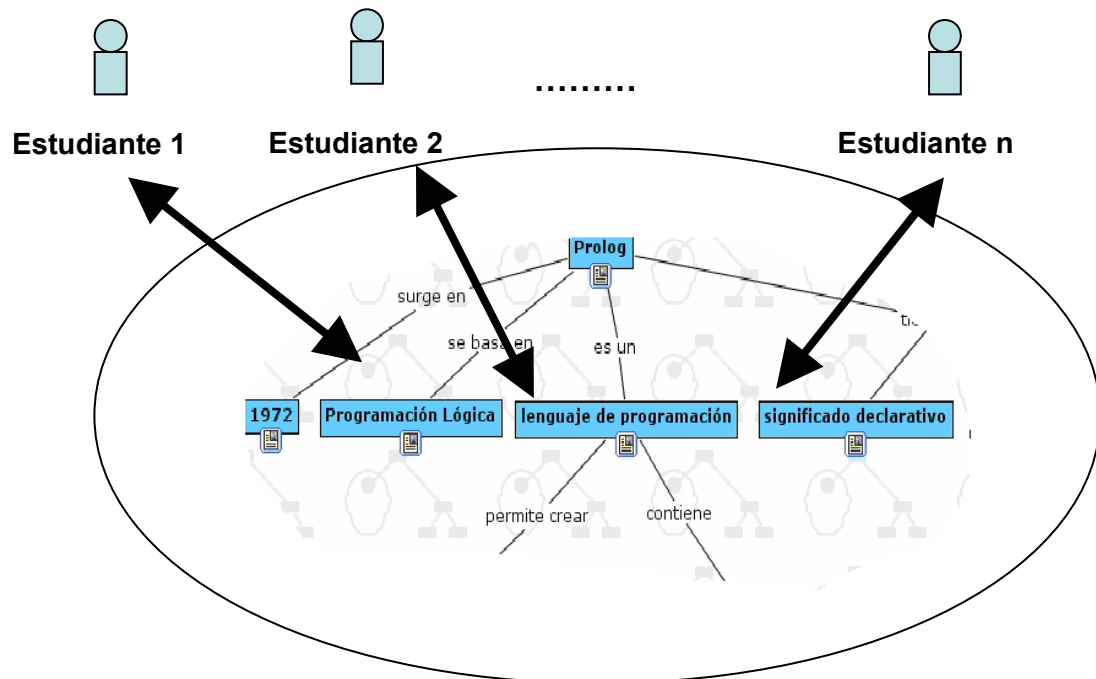


Figura 3.2 Relación bidireccional entre los estudiantes y el ambiente.

Por otro lado, este intercambio, no sólo hace al estudiante protagonista de su propio aprendizaje sino un tanto responsable del aprendizaje del resto de sus compañeros de año y de los de cursos venideros. Lo anterior contribuye a fomentar la responsabilidad, laboriosidad y otros valores que deben caracterizar a un nuevo profesional.

La experiencia ha resultado enriquecedora y ha dado lugar a otros dos posibles ejercicios:

1. Proponer a los estudiantes que continúen agregando hechos y reglas a las bases de conocimiento mencionadas con anterioridad, con lo que se les invita a profundizar en todos estos temas también necesarios para el logro de una cultura general integral y con lo que se contribuye a la formación de habilidades investigativas.
2. La invitación, casi espontánea, para que los estudiantes de cursos posteriores y de otras universidades cubanas busquen otras y mejores soluciones a las aquí planteadas. En particular se han hecho relaciones con la UCLV para realizar estas tareas y expandir el alcance de la herramienta.

Utilización de APA-Prolog a partir de los conceptos tratados en el aula

Seguidamente se expone la forma en que se ha utilizado el ambiente en las actividades de autoaprendizaje correspondientes a las clases de Programación Descriptiva en el CUSS, teniendo en cuenta que la creatividad de los docentes juega un importante papel en la organización de esta parte del proceso de enseñanza-aprendizaje.

En la primera clase del curso se mostró a los estudiantes el ambiente y se les explicó como navegar a través de él. Luego los recursos se fueron orientando en la misma medida en que se fueron introduciendo los temas afines. Al finalizar cada actividad presencial el docente entregó una guía para la navegación, la cual incluye:

- Objetivos a vencer durante la navegación.
- Habilidades a alcanzar.
- Conceptos esenciales a visitar.
- Orientaciones para la navegación.
- Actividades directas a realizar (Preguntas a contestar, ejercicios a resolver, tablas, resúmenes, etc.).

En la clase siguiente el docente controló la realización de las actividades realizadas y la calidad de las mismas. Esta rutina se llevó a cabo en la casi totalidad de las actividades del semestre.

3.2 Metodología para la creación de ambientes de enseñanza-aprendizaje usando mapas conceptuales que toman en cuenta los conocimientos previos del estudiante

Como resultado de esta investigación se obtuvo una metodología que pretende homogenizar los procedimientos para desarrollar sistemas de enseñanza-aprendizaje que se apoyan en el conocimiento previo de los estudiantes.

3.2.1 Generalidades

Para diseñar e implementar ambientes de enseñanza-aprendizaje que permitan tomar en cuenta los conocimientos previos del estudiante se deben seguir algunos pasos que son independientes del lenguaje de programación o las herramientas utilizadas. La experiencia propia y la bibliografía consultada han permitido resumirlos en los siguientes puntos:

- Si el dominio del conocimiento a representar es muy extenso se deben construir varios mapas conceptuales pequeños con significación propia, o sea, no se debe elaborar un gran mapa conceptual, ya que esta forma de trabajo trae aparejadas algunas desventajas, entre las que cabe citar:
 - a) El estudiante se abruma cuando ve que tiene tantos caminos por recorrer.
 - b) Resulta difícil detectar conceptos, palabras enlaces o relaciones erróneas.
 - c) No resulta fácil la inclusión de nuevos conceptos.
- Se debe seleccionar cuidadosamente la herramienta con la que se van a implementar los mapas conceptuales previamente diseñados o se puede crear una propia. Se recomienda el CmapTools por lo que esta metodología supone que se trabaja con dicho software.
- Para facilitar la colocación posterior en un servidor, se sugiere que todos los recursos que se adjuntarán a los conceptos de los mapas se coloquen en una carpeta principal que contendrá varias subcarpetas clasificadas de acuerdo al tipo de recurso que contienen. En este aspecto debe señalarse que se pueden emplear diferentes servidores (bases de datos, Web, etc.) por lo que la distribución de carpetas puede

estar sujeta a la propia configuración de los servidores necesarios y es posible que existan varias carpetas separadas que no tengan necesariamente relaciones de ascendencia-descendencia.

- El ambiente debe facilitar tanto la navegación controlada por los agentes inteligentes como la navegación libre para aquellos usuarios que así lo prefieran.
- La interfaz de usuario del producto final debe ser cómoda de manera que no se requiera un largo período de adiestramiento para poder usarlo.

3.2.2 Metodología

Se presentan a continuación las ideas generales que se proponen para el desarrollo de ambientes de enseñanza-aprendizaje basados en mapas conceptuales con posibilidades de tener en cuenta los conocimientos previos del estudiante.

Diseño del diagnóstico del estudiante.

Los sistemas desarrollados con esta intención precisan de una representación del estado actual del conocimiento del aprendiz que se dispone a utilizar el ambiente. Es decir, la evaluación, en las categorías de: bien, regular o mal, del nivel de conocimiento que, acerca de los conceptos principales, posee el usuario. Esta información es decisiva para personalizar el ambiente y para lograr que sea lo más fiel posible se recomienda:

Paso 1: Determinar qué conceptos serán abordados en el mapa conceptual.

Paso 2: Hacer un diagnóstico del estado actual de conocimiento del estudiante sobre cada uno de los conceptos determinados en el paso anterior.

Paso 3: Guardar ese diagnóstico inicial en una base de datos.

Paso 4: Mantener el diagnóstico abierto al estudiante, de forma tal que este pueda modificarlo siempre que lo entienda necesario.

Diseño de los recursos que contienen el conocimiento.

El núcleo del conocimiento a transmitir lo constituyen los mapas conceptuales que servirán de andamiaje estructural para que los usuarios naveguen sobre ellos.

Para conseguir que los estudiantes no se queden solo en la apreciación de los conceptos y sus relaciones sino que puedan profundizar en ellos se distribuye el resto del contenido en otros recursos informáticos como pueden ser textos explicativos, sitios Web, presentaciones, simuladores, tutoriales, entrenadores, evaluadores y otros según requiera la materia en cuestión. Todo este proceso lo podemos sintetizar en los siguientes pasos:

Paso 1: Determinar los conceptos principales del dominio del conocimiento que se quieren representar.

Paso 2: Seleccionar cuidadosamente las palabras enlaces que relacionarán los conceptos, sin olvidar la relevancia que ellas tienen para que el mapa exprese realmente lo que se quiere. Es importante la precisión debido a que pueden existir infinitas relaciones entre dos conceptos.

Paso 3: Construir una primera versión del mapa conceptual jerárquico. (Esta primera versión casi nunca es la última.)

Paso 4: Evaluar la calidad final de ese mapa conceptual, atendiendo a los siguientes criterios: (En la literatura revisada aparecen muchos pero la autora considera que, al menos estos, no se pueden obviar)

1. Jerarquía de conceptos. Es decir, que cada concepto inferior dependa del superior (en el contexto de lo que ha sido planteado).
2. Cantidad y calidad de los conceptos.
3. Buena relación de los significados entre dos conceptos conectados por la línea indicada y las palabras apropiadas.
4. Que exista una conexión significativa entre un segmento de la jerarquía y el otro, es decir, deben existir ligámenes significativos y válidos entre conceptos.
5. Claridad. Esta se logra evitando que se crucen líneas innecesariamente, que se solapen conceptos y que se empleen iconos similares para recursos diferentes.

Paso 5: Construir versiones sucesivas del mapa conceptual hasta llegar a la definitiva que será aquella donde se satisfagan todos los criterios anteriores.

Paso 6: Someterla al criterio de expertos en el contenido.

Paso 7: Si el mapa resultante es muy grande dividirlo en otros más pequeños con sentido propio.

Paso 8: Implementar cada uno de esos mapas en CmapTools u otra herramienta similar.

Paso 9: Si el paso 7 se realizó, enlazar adecuadamente los mapas que se derivaron del gran mapa original.

Paso 10: Colocar en la carpeta general todos los recursos informáticos que se deseen añadir a los mapas conceptuales. Estos recursos pueden ser de cualquier naturaleza, es decir, por ejemplo: .pdf, .doc, .xls, .ppt, .html, .avi y otros. La carpeta debe dividirse en subcarpetas organizadas por categorías.

No es necesario tener disponibles todos los recursos en este paso, pero sí debe definirse la organización general para que después puedan incluirse fácil y progresivamente.

Se recomienda que los recursos sean fácilmente identificados por sus nombres y estén en letra minúscula para evitar problemas en aquellas plataformas que sean sensibles a ese aspecto. Los nombres de los recursos no deben tener demasiados caracteres debido a que los caminos con cadenas de caracteres grandes pueden ser no validos sobre determinados soportes de información.

Paso 11: Someter este primer prototipo del ambiente a la consideración de los estudiantes y recoger sus criterios mediante encuestas, entrevistas u otra técnica y repetir este paso mientras los estudiantes tengan algo que señalar.

Paso 12: Cuando ya el mapa esté terminado se coloca en el servidor para que pueda ser visto desde cualquier otro lugar. Se recomienda usar un servidor GNU Linux.

Diseño de los agentes para la adaptación.

Los pasos descritos anteriormente permiten obtener un ambiente formado por un conjunto de mapas conceptuales con diversos recursos asociados a sus conceptos sobre los cuales los estudiantes pueden navegar libremente. Sin embargo, todos los estudiantes reciben los mismos contenidos, se le presentan los mismos ejercicios propuestos o resueltos, etc.

Una plataforma mejorada, debe tomar en cuenta el estado de conocimiento del estudiante para ofrecerle un tratamiento diferenciado como se describió previamente.

Para poder tomar en cuenta el estado de conocimiento, se pueden usar diferentes técnicas de Inteligencia Artificial, en este trabajo se recomienda el empleo de agentes inteligentes, en particular se concibe un sistema multiagente, en el cual cada agente tendrá una responsabilidad específica. Los agentes Tivo, Teo, Tica y Eva tendrán acciones propias a desarrollar de acuerdo al algoritmo general que se muestra en la figura 3.3.

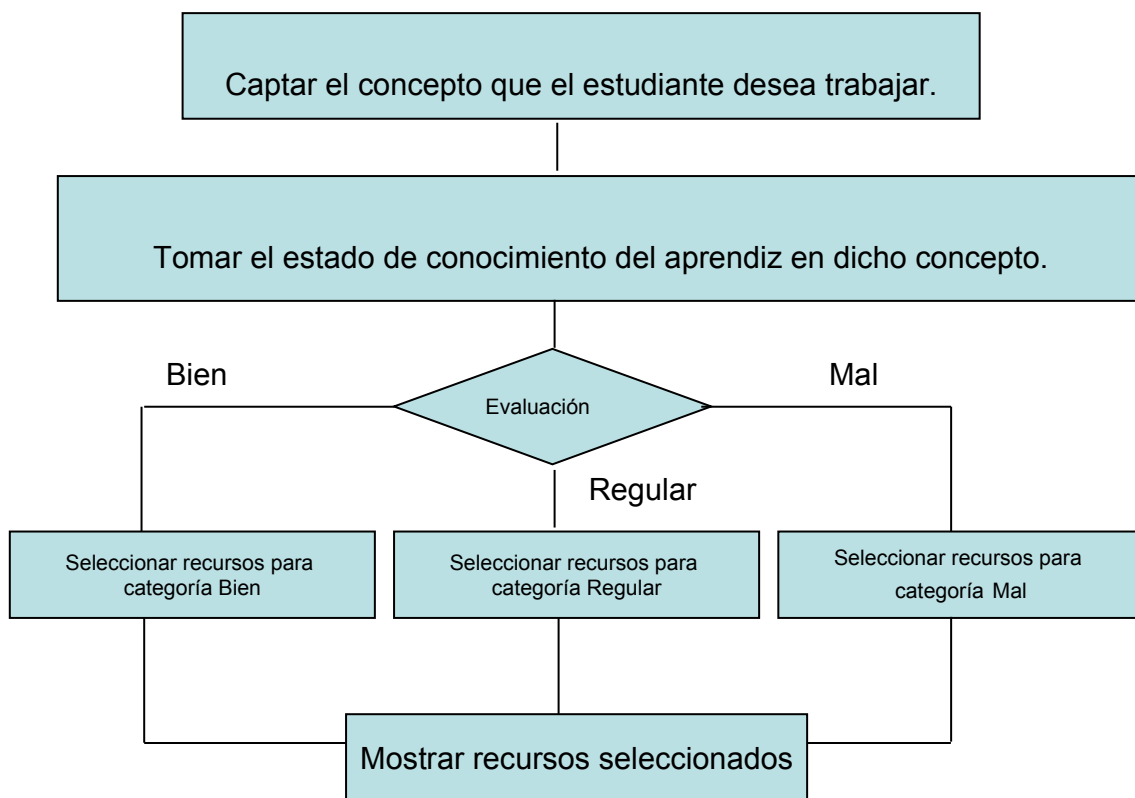


Figura 3.3 Algoritmo general de trabajo de los agentes Tivo, Teo, Tica y Eva.

Consideración final

Aunque se recomienda CmapTools para hacer el ambiente, pudiera usarse cualquier otra herramienta, la cual debe proporcionar algunas libertades que permitan incluir la programación de los agentes inteligentes.

El hecho de que CmapTools no sea un software de código abierto, imposibilita incluir dentro de él la programación mencionada. Existen algunas herramientas con este mismo fin que son libres y por tanto se tiene acceso al código, pero tienen otras limitaciones para hacer sistemas de enseñanza amigables y poderosos.

No obstante, el señalamiento anterior CmapTools es gratuito y posee una licencia que ofrece facilidades para aplicaciones con fines docentes. Para poder introducir los agentes en el ambiente es necesario exportar todos los mapas confeccionados a formato Web, el propio CmapTools cuenta con una opción que lo permite, una vez hecha esta labor se pueden programar los agentes en algún lenguaje de programación adecuado que en este caso fue PHP.

3.2.3 Empleo de la metodología en otras áreas de conocimiento.

Los lineamientos especificados en la metodología propuesta fueron seguidos estrictamente en el desarrollo de APA-Prolog.

Como una forma de validar la posible extensión de la metodología hacia otras áreas del conocimiento, se desarrollaron dos aplicaciones para la enseñanza de otras materias: Estadística Matemática I e Historia de la Psicología. Estas aplicaciones están en la fase de validación de los mapas y por ello no tienen incorporadas, aún, las posibilidades de adaptación dadas por los agentes inteligentes.

Asignatura Estadística Matemática I

Para apoyar el proceso de enseñanza-aprendizaje de la asignatura Estadística Matemática I, correspondiente al plan de estudios C de la carrera Ingeniería Informática se diseñó un sitio Web formado por mapas conceptuales de cada uno de los temas de la asignatura: “Estadística Descriptiva”, “Probabilidades”, “Muestreo y estimación” y “Pruebas de Hipótesis”. Este sitio se confeccionó durante el curso 2007-2008 y los estudiantes comenzaron a utilizarlo en el curso 2008-2009.

La herramienta potencia el estudio individual, proporciona recursos tales como manuales sobre software estadísticos, presentaciones y diversos materiales tomados de otras universidades del país donde se imparte la carrera desde hace varios años. Cada tema tiene,

al menos, un mapa. Cada uno de ellos permite acceder de forma organizada a las conferencias, clases prácticas, y otros medios elaborados por los docentes. Cada concepto tiene su explicación y, en la mayoría de los casos ejemplos, y ejercicios propuestos.

Asignatura Historia de la Psicología

La asignatura Historia de la Psicología se imparte en el décimo semestre de la carrera Licenciatura en Psicología, en la modalidad semipresencial. El programa de estudio consta de 16 horas de encuentros y 48 horas de estudio independiente para un total de 64 horas.

En este caso, se orientó, en el primer encuentro, el uso de la parte 2 de la metodología propuesta anteriormente con la intención de que los estudiantes fueran construyendo sus mapas conceptuales con CmapTools y los fueran enriqueciendo en la misma medida en que se apropiaban de los conocimientos. Esta estrategia hizo posible que el trabajo de control se basara en la exposición de momentos del desarrollo histórico de la psicología como ciencia según el período que le correspondiera de forma tal que se obtuviera un resumen del período desde el nacimiento de la humanidad hasta el surgimiento de la psicología como ciencia.

Se espera, al finalizar el semestre, recibir representaciones de todos los períodos para usarlas en próximos cursos escolares como parte del plan de desarrollo de la asignatura y los mejores mapas quedarán como material de apoyo a la docencia.

3.3 Validación estadística de la efectividad de APA-Prolog

La validación de la efectividad del sistema se hizo mediante encuestas aplicadas a estudiantes de cuarto año de la carrera “Ingeniería Informática” que cursaron la asignatura Programación Descriptiva durante el curso 2008-2009 en el Centro Universitario de Sancti Spiritus “José Martí”.

La encuesta estuvo compuesta por seis preguntas que pueden apreciarse en el anexo 1. Las respuestas a las mencionadas interrogantes se detallan en el anexo 2.

Para validar el comportamiento del sistema, los resultados de la encuesta fueron sometidos a un análisis de frecuencia para conocer el comportamiento univariado de las respuestas y un análisis de cluster jerárquico para conocer su comportamiento multivariado,

calculándose la moda y mediana para cada uno de los grupos que se formaron. Se utilizó el paquete SPSS Versión 15.0.1 (Statistical Package for the Social Sciences), los resultados del análisis de frecuencia se aprecian en el anexo 3 (tablas 1, 2, 3, 4 y 5).

En la tabla 1 se puede apreciar que 23 estudiantes (de un total de 25 que se encuestaron) consideran que la forma de representar el conocimiento a través de mapas conceptuales -definitivamente sí le ha facilitado comprender los conceptos claves de la Programación Lógica- lo que representan el 92 % de los encuestados.

Por su parte, la tabla 2 deja ver que el 76 % de la muestra opina que APA-Prolog , le ha ayudado mucho a transformar las especificaciones de un problema dado en hechos, reglas y preguntas.

De apreciable, evalúan 18 estudiantes (72 %) la influencia que ha tenido el ambiente en su capacidad de expresar claramente un algoritmo empleando el estilo descriptivo, mientras el 48 % evalúa de medio alto la elevación de su nivel de conocimiento acerca de la Programación Lógica a partir de su interacción con el ambiente. (Tabla 4)

La tabla 5 permite observar que el 100 % de los estudiantes considera que, ambientes similares a APA-Prolog facilitarían el proceso de enseñanza-aprendizaje de otros temas de otras asignaturas.

Las respuestas a la pregunta 4 del anexo 1: ¿Qué aspectos te gustaría cambiar en el ambiente? Se procesó con el núcleo central de las representaciones sociales²³ y resultaron los aspectos más frecuentes y más importantes: la necesidad de una tabla o índice que indique en qué mapa se encuentra cada concepto y el deseo de la presencia de elementos animados y de mayor colorido en el ambiente.

En todas las preguntas se observan respuestas favorables y como puede apreciarse en el anexo 2 en las preguntas 3 y 5 es donde más variabilidad de criterios hay.

Al realizar el análisis de clúster jerárquico (corte 1) se obtiene el dendrograma que se muestra en la figura 3.4 con un primer corte a distancia un poco mayor que 5 se forman tres grupos de tamaño 1, 14 y 10.

²³ <http://gema.univalle.edu.co/hvida/hvmaria.html>

La moda y la mediana para cada pregunta del cuestionario se muestran en la tabla 1 del anexo 4.

Si el corte se realiza a una distancia mayor (corte 2) se forman dos grupos: Uno de un alumno y otro del resto (24). La moda y la mediana para cada pregunta del cuestionario se muestran en la tabla 2 del anexo 3.

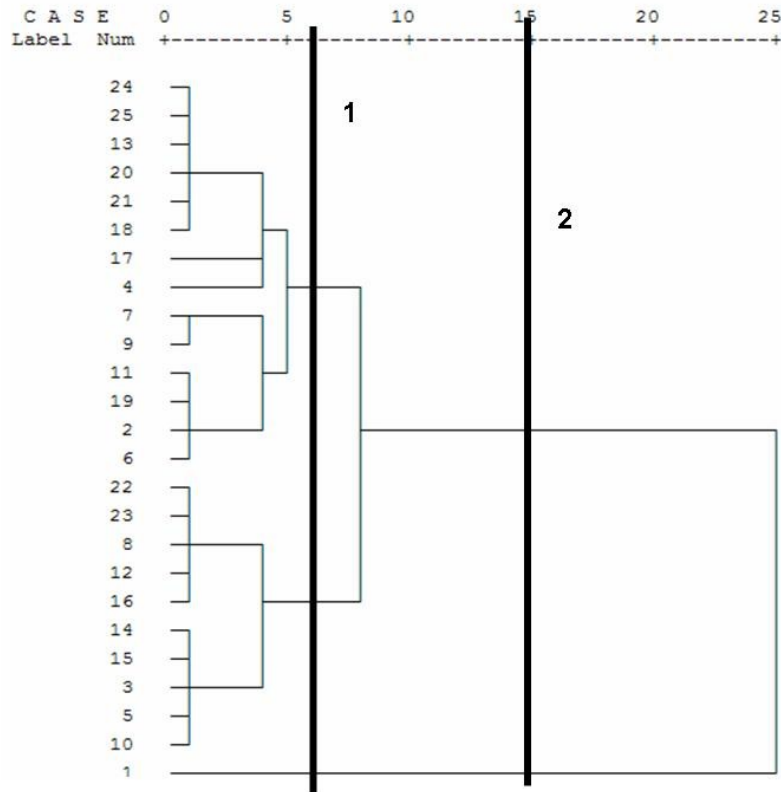


Figura 3.4 Dendrograma correspondiente al análisis de cluster realizado. Caso 1

Como se puede apreciar la diferencia entre grupos está en las preguntas 3 y 5, que además son las preguntas que en el análisis de frecuencia tienen más variabilidad en las respuestas.

3.4 Análisis de los resultados valorados por expertos

Se utilizó el método Delphi para la selección y procesamiento de las encuestas a los expertos. En el Anexo 5 aparecen los aspectos que se tuvieron en cuenta para la selección de los expertos, los que emitieron sus criterios en cuanto al ambiente que se propone.

Para elegir la muestra se tuvo en cuenta a expertos de más de diez años de experiencia como profesionales y profesores de las disciplinas Programación de Computadoras e Inteligencia Artificial, con dominio de la metodología de la enseñanza de la programación y en el diseño de software educativo. De los 35 expertos inicialmente previstos, 30 cumplieron con los requisitos planteados en la encuesta, de ellos el 80.0% con un nivel de competencia Alta y el 20.0% Media, todos tienen categorías docentes principales, 46.7 % son Auxiliares, 53.3% Titulares, 26.7 % son Máster y 73.3% Doctores (Anexo 6).

A los seleccionados se les mostró en detalles el ambiente propuesto, y se les aplicó una encuesta con los elementos a evaluar (Anexo 7).

3.4.1 Procesamientos estadísticos utilizados

En esta parte del trabajo se hace un análisis descriptivo de datos, utilizando en general frecuencias para variables dicotómicas y proporciones de respuestas positivas, como es el caso de la selección de los recursos y la asignación de un orden de importancia a cada uno. En estas preguntas en particular se utiliza un test de Cochran para distinguir cuáles son los recursos que de acuerdo a la proporción fueron significativamente más marcados y cuáles menos. Pero además, como el cuestionario incluía una pregunta sobre el ordenamiento de la importancia de los mismos, se aplicó un test de concordancia de Kendall para medir el acuerdo entre los expertos.

3.4.2 Valoraciones de los expertos

La Tabla 3.1 muestra los resultados de la encuesta a los expertos, sobre la forma en que considera que este ambiente permite elevar los niveles de asimilación de los contenidos relacionados con la Programación Lógica. Puede observarse que el 66,7 % opina que es muy adecuada y el restante 33,3 % que es adecuada.

Tabla 3.1 Respuesta de los expertos a la pregunta 1 del anexo 7.

Respuestas	Frecuencia	Por ciento
Adecuado	10	33,3
Muy adecuado	20	66,7
Total	30	100,0

En la tabla 3.2 se aprecia que el 100 % de los expertos considera que existe, entre todos los recursos disponibles en el ambiente, la armonía necesaria para propiciar que los estudiantes se apropien de los contenidos de la asignatura.

Tabla 3.2 Respuesta de los expertos a la pregunta 2 del anexo 7.

Posibles Respuestas	Frecuencia	Por ciento
Nada	0	0
Poco	0	0
Mucho	30	100,0

La posibilidad que tiene el ambiente de tomar en cuenta los conocimientos previos de los estudiantes fue considerada por el 53,3 % de los expertos como muy buena, mientras que el 46,7 % opina que es buena. (Tabla 3.3)

Tabla 3.3 Respuesta de los expertos a la pregunta 3 del anexo 7.

	Frecuencia	Por ciento
Buena	14	46,7
Muy buena	16	53,3
Total	30	100,0

La cantidad de marcas que recibió cada recurso se resume en la figura 3.5, donde se puede apreciar que excepto el evaluador que no fue seleccionado por uno de los expertos los demás recursos fueron marcados por todos. El test de Cochran confirma que no hay diferencias significativas entre las marcaciones de los ocho recursos.

La Figura 3.6 complementa la información anterior, a partir del orden de importancia que explícitamente dieron los expertos a cada uno de los recursos.

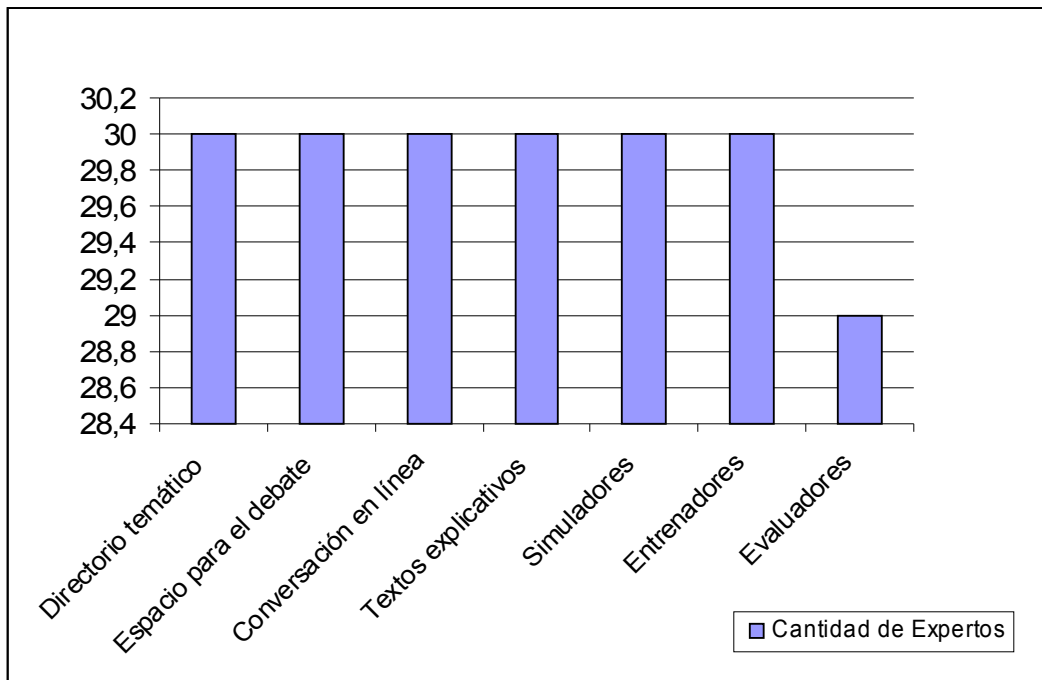


Figura 3.5 Cantidad de marcas que recibió cada recurso.

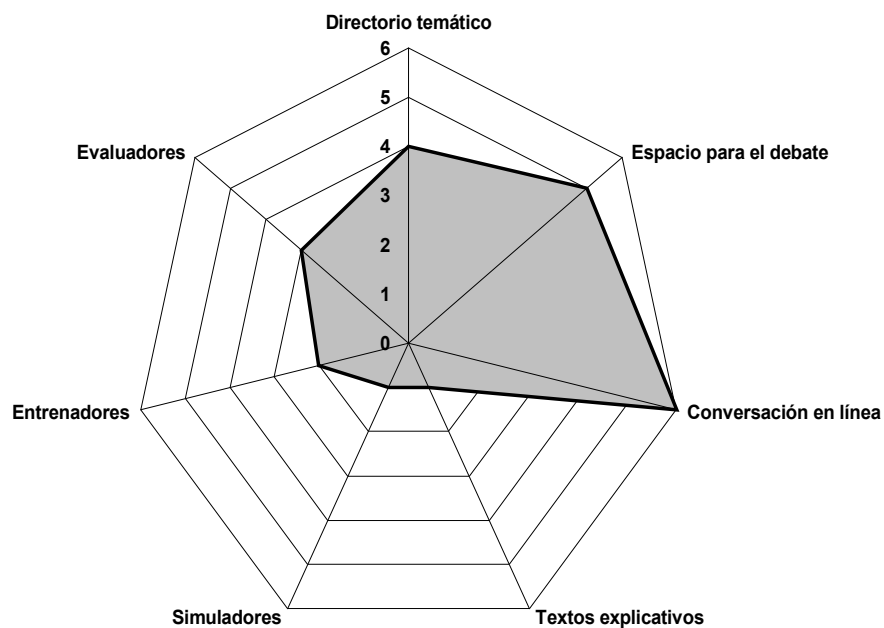


Figura 3.6 Orden dado a los recursos por los expertos.

Para determinar hasta qué punto hay una concordancia en el orden asignado por los expertos se aplica un test de concordancia de Kendall. El acuerdo medido por el coeficiente de concordancia no es muy alto ($W=0.433$) pero es bastante diferente de cero (que sería un total desacuerdo).

Los rangos medios de las respuestas están significativamente bien marcados (significación 0.000) y por tanto se puede llegar a un orden, consenso de la importancia asignada a los recursos a partir de ellos (un rango medio menor significa una importancia promedio mayor). Así, resultan evaluados de más importantes los textos explicativos, los simuladores, los entrenadores y los evaluadores y menos importantes: el directorio temático, el espacio para el debate y la conversación en línea. (Anexo 8)

Se destaca el hecho de que 11 de los 30 expertos otorgaron el mismo orden de prioridad a todos los recursos lo que deja ver que actúan como un sistema y que se complementan unos con otros.

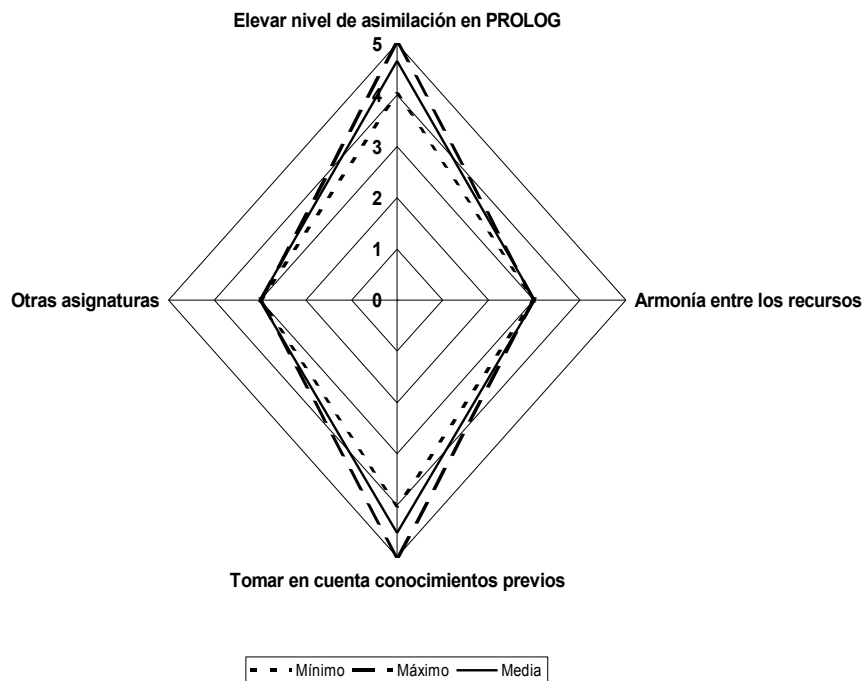


Figura 3.7 Resumen de la valoración de los expertos

Por último, al preguntarles a los expertos si ambientes similares a APA-Prolog facilitarían el proceso de enseñanza-aprendizaje de otros temas de otras asignaturas el 100 % respondió afirmativamente.

La figura 3.7 resume las valoraciones de los expertos sobre los aspectos tratados en las preguntas 1, 2, 3 y 5 de la encuesta aplicada.

3.5 Conclusiones Parciales

Para elevar la calidad de la enseñanza de la Programación Lógica se hace necesaria la existencia de medios o herramientas que logren dar significado y relevancia al nuevo contenido partiendo de los conocimientos previos que poseen los estudiantes, al mismo tiempo que despierten su interés por aprender.

Se obtiene una metodología que detalla los lineamientos generales para desarrollar sistemas de enseñanza inteligentes basados en Web que se aplicó al sistema descrito en el capítulo 2 y se extendió a otras dos aplicaciones para validar su generalidad.

Con el ambiente desarrollado se logra una participación activa de los educandos en el proceso de enseñanza-aprendizaje de la Programación Lógica, así como la adquisición de habilidades y competencias profesionales propias de un ingeniero informático.

CONCLUSIONES Y RECOMENDACIONES

1. Se evidenció, a través del análisis bibliográfico que, el desarrollo de investigaciones donde se emplean los mapas conceptuales, las TIC y los agentes inteligentes da lugar a herramientas informáticas que apoyan los procesos de enseñanza-aprendizaje en diferentes entornos. La enseñanza de la Programación Lógica, debido a complejidades propias, precisa de medios y herramientas basados en individualidades cognitivas.
2. Se obtiene el sistema APA-Prolog, basado en mapas conceptuales, que integra contenidos de Programación Lógica en un ambiente de navegación que permite relacionar los conceptos fundamentales de ese paradigma.
3. Se elabora un conjunto de herramientas que auxilian el proceso de enseñanza-aprendizaje de la Programación Lógica, las cuales se insertan armónicamente, como recursos informáticos, a diferentes nodos de APA-Prolog.
4. El diseño de un conjunto de agentes inteligentes permitió dotar a APA-Prolog de capacidad para elegir los contenidos a mostrar tomando en cuenta los conocimientos individuales de los estudiantes.
5. Los instrumentos aplicados a los estudiantes muestran la efectividad de APA-Prolog medida en: comprensión de conceptos claves y capacidad para interpretar las especificaciones de un problema reflejándolas en un programa. Por otra parte los instrumentos aplicados a los expertos dan fe de la calidad del sistema.

RECOMENDACIONES

1. Implementar ambientes similares a APA-Prolog para apoyar el proceso de enseñanza-aprendizaje de otras asignaturas.
2. Ampliar las posibilidades de adaptación de APA-Prolog, de forma que sea capaz de tomar en cuenta otras necesidades o características de los estudiantes.

REFERENCIAS BIBLIOGRÁFICAS

- Almeida, S. (2007). Metodología para la gestión del conocimiento en Ciencias Básicas Biomédicas con el empleo de las Tecnologías de la Información y las Comunicaciones. Tesis de Doctorado. Universidad de Matanzas "Camilo Cienfuegos", Matanzas.
- Almeida, S., Febles, J. y Bolaños, O. (1997). Evolución de la enseñanza asistida por computadoras. *Revista de Educación Médica Superior* 11(1), 31-38.
- Alonso, F. (2005). The Synthesis Stage in the Software Agent Development Process. CEEMAS, LNAI 3690. Springer-Verlag Berlin Heidelberg 2005, 193 – 202.
- Baray, C. y Wagner, K. (1999). ¿Where Do Intelligent Agents Come From? . ACM, Volumen 5(Número 4), ISSN: 1528-4972 p. 1524-1528.
- Bauer, B., y Müller, J. (2004). Using UML in the Context of Agent-Oriented Software Engineering: State of the Art. AOSE, LNCS 2935. Springer-Verlag Berlin Heidelberg 2005, 1–24.
- Berlanga, A. J. y García, F. J. (2004). Sistemas Hipermedia Adaptativos en el ámbito de la Educación. Departamento de Informática y Automática. Departamento de Informática y Automática. Universidad de Salamanca. España.
- Bratko, I. (1990). Prolog programming for artificial intelligence. Addison-Wesley Publishers. Second Edition. ISBN 0-201-41606-9.
- Brusilovsky, P. (1996). Methods and techniques of adaptive hipermedia. *Journal of User Modeling and User Adapted Interaction*, Vol. 6 (2-3), 87-129.
- Busquets, D. (2008). A Multiagent Approach to Qualitative Navigation in Robotics. Tesis de Doctorado. Universidad Politécnica de Cataluña, España.
- Casali, A. (2008). A Tourism Recommender Agent: from theory to practice. *Inteligencia Artificial* , 40, 23-38.
- Choua, C. (2002). Redefining the learning companion: the past, present, future of educational agents. *Computer & Education*, 40 (3), 255-269.
- Cuena, J. y García, A. (1992). Sistemas basados en el conocimiento para coordinación de trabajo cooperativo. 3er Congreso Iberoamericano de Anteligencia Artificial. La Habana. Febrero 1992, 321-335.

- Cuevas, S. (2007). Propuesta de aplicación de los mapas conceptuales en un modelo pedagógico semipresencial. *Revista Hibernoamericana de Educación* (ISSN: 1681-5853)
- Christiansen, H. (2005). HYPROLOG: A New Logic Programming Language with Assumptions and Abduction. *ICLP 2005, LNCS 3668, Springer-Verlag Berlin Heidelberg 2005*, 159 -173.
- Das, S. (2008). PROLOG: A multiagent toolkit for logistics planning in netcentric environments. *11th International Conference on Information Fusion, 2008*, 1-8.
- Dávila, S. y Martínez, G. (2000). Curso Taller “Mapas Conceptuales, en busca del aprendizaje significativo”. Universidad de Champagnat. San Luis de Potosí. México.
- De Bra, P., Brusilovsky, P. y Hoben, G. (1999). Adaptive hypermedia: from systems to framework. *ACM Computing Surveys*, 31 (4).
- Demazeau, Y. (2000). Foundations of Multi-agent System. 2nd European Agent Systems Summer School. EASSS2000. Saarbrucken. Alemania.
- Demoen, J. S. y. T. S. y. B. (2005). Guard and Continuation Optimization for Occurrence Representations of CHR. *ICLP 2005, LNCS 3668, Springer-Verlag Berlin Heidelberg 2005*, 83-97.
- Díaz, A. (2006). Usando el entorno Web para la Universalización de la enseñanza de la Matemática. Trabajo de Diploma. Licenciatura en Ciencias de la Computación. Universidad Central "Marta Abreu" de Las Villas. .
- Díaz, D. (2009). Sistema Multiagente para el diagnóstico presuntivo de enfermedades ginecológicas. Tesis para optar por el Título de Ingeniería en Informática. Universidad de Cienfuegos "Carlos Rafael Rodríguez", Cuba.
- Dragone, M., Holz, T., y Duffy, B. (2005). Social Situated Agents in Virtual, Real and Mixed Reality Environments. *IVA, LNAI 3661. Springer-Verlag Berlin Heidelberg 2005*, p. 166 – 177.
- Elliot, C., Rickel, J., y Lester, J. (1997). Integrating affective computing into animated tutoring agents. *IJCAI workshop on Animated Interface Agents. Nagoya. Japón.*
- Estrada, V., y Flebes, J. (2002). Mapas Conceptuales. Universidad de Guadalajara Aplicaciones de la Inteligencia Artificial. Vol. 3, p. 400. México.

- Ferber, J. (1999). *Multi-agent Systems: An introduction to distributed Artificial Intelligence*. Addison Wesley. England.
- Fernández, K. (2005). *Agentes Inteligentes para el diagnóstico de patologías ginecológicas*. Tesis de Maestría. Universidad Central de Las Villas, Villa Clara, Cuba.
- Fischer, G. (2000). *User Modeling in Human-Computer Interaction*. Contribution to the 10th Anniversary Issue of the Journal "User Modeling and User-Adapted Interaction (UMUAI)".
- García, A. (2007). *Agentes Inteligentes para la atención a la diversidad de estudiantes en la modalidad de Educación a Distancia a través de los sistemas informáticos*. Tesis de Maestría.
- Gillies, M., Robertson, D. y Ballin, D. (2005). *Direct Manipulation Like Tools for Designing Intelligent Virtual Agents*. IVA, LNAI 3661. Springer-Verlag Berlin Heidelberg 2005, p. 430–441.
- Giraffa, L. y Viccari, R. (1998). *The use of agents techniques on intelligent tutoring systems* International Conference of the Chilean Society of Computer Science. Proceedings IEEE Computer Society.
- González, C. (2004). *Sistemas Inteligentes en la Educación: Una revisión de las líneas de investigación y aplicaciones actuales*. Revista Electrónica de Investigación y Evaluación Educativa, Vol. 10, No 1.
- González, M. (2007). *Simulador del trabajo con Listas en Prolog*. Tesis de Maestría. Centro Universitario de Sancti Spiritus "José Martí Pérez".
- González, P. y Barreiro, A. (1993). *Arquitectura orientada a objetos para sistemas expertos distribuidos*. X Reunión Nacional de Inteligencia Artificial. Memorias. México. 1993.
- Grimaldo, F., Lozano, M., Barber, F. y Orduna, J. (2005). *Integrating Social Skills in Task-Oriented 3D IVA*. IVA, LNAI 2650. Springer-Verlag Berlin Heidelberg, 358–370.
- Grimón, F. (2008). *Modelo para la gestión de dominios de contenido en sistemas hipertexto adaptativos aplicados a entornos de Educación Superior*. Universidad Politécnica de Cataluña, Barcelona. España.
- Hernansáez, A. (2005). *Asistencia personalizada a la Minería de Datos mediante Agentes Inteligentes*. Departamento de Ingeniería de la Información y las Comunicaciones.

- Universidad de Murcia. Rev. Intelligent Multimedia Information Retrieval in the Net. Universitat Pompeu Fabra.
- Hjelmlom, M. (2009). Agent and Multi-Agent Systems: Technologies and Applications. Third KES International Symposium, KES-AMSTA 2009 Uppsala, Sweden, June 2009. Proceedings. LNAI 5559. Springer Berlin Heidelberg, 110-119.
- Ibarra, S. (2008). A formalization for multi-agent decision support in cooperative environments. A framework for situated agents. Tesis de Doctorado. Universidad de Girona, España.
- Innocenti, B. (2008). A multiagent architecture with distributed coordination for an autonomous robot. Tesis de Doctorado. Universidad de Girona, España.
- Jiyong, M., Jie, Y. y Cole, R. (2002). CU Animate Tools for Enabling Conversations with Animated Characters., International Conference on Spoken Language Processing. (Vol. Vol 1, pp. 197-200). Denver, CO USA.
- Jonker, C., Klusch, M. y Treur, J. (2000). Design of Collaborative Information Agents. CIA, LNAI 1860. Springer-Verlag Berlin Heidelberg 2000, 262-283.
- Julian, V. y Botti, V. (2000). Agentes Inteligentes: el siguiente paso en la Inteligencia Artificial. Universidad Politécnica de Valencia. Departamento de Sistemas Informáticos y Computación. Rev Novatica. Edición Digital Especial 25 Aniversario. España.
- Kopp, S., Gesellensetter, L. y Krämer, N. (2005). A Conversational Agent as Museum Guide – Design and Evaluation of a Real-World Application. IVA, LNAI 3661. Springer-Verlag Berlin Heidelberg 2005, p. 329 – 343.
- Krämer, N. C. (2005). Social Communicative Effects of a Virtual Program Guide. IVA, LNAI 3661. Springer-Verlag Berlin Heidelberg 2005, p. 442 – 453.
- Lezcano, M. (1998). Ambientes de aprendizaje por descubrimiento para la disciplina Inteligencia Artificial. Tesis de Doctorado. Universidad Central "Marta Abreu" de Las Villas, Santa Clara. Cuba.
- Lezcano, M. y Valdés, G. (1998). Algunas Experiencias en la Utilización de Sistemas de EAC para la Enseñanza de la Inteligencia Artificial. Revista Divulgaciones Matemáticas, Volumen 6 No. 2, 165-182.

- López, M. (2007). Modelo pedagógico del tutor en la enseñanza-aprendizaje del inglés a distancia con el aprovechamiento de las TIC en el postgrado. Tesis de Doctorado. Instituto Superior Pedagógico “Enrique José Varona” Facultad de Lenguas Extranjeras, Ciudad de La Habana. Cuba.
- Maatman, R., Gratch, J. y Marsella, S. (2005). Natural Behavior of a Listening Agent. IVA, LNAI 3661. Springer-Verlag Berlin Heidelberg 2005, 25 – 36.
- Maes, P. (1994). Modeling Adaptative Autonomous Agents. MIT Media-Laboratory, Cambridge. Artificial Life Journal. Vol. 1, No 1. p. 35-162.
- Martin, A., Gregory, M. y Bradley, J. (2005). Maintaining the Identity of Dynamically Embodied Agents. IVA, LNAI 3661. Springer-Verlag Berlin Heidelberg 2005, 454–465.
- Martínez, A. (2007). Simulador de Bactracking. Tesis de Maestría. Centro Universitario de Sancti Spíritus "José Martí".
- Medina, D. (2007). Redes Bayesianas y Mapas Conceptuales en la elaboración de Sistemas de Enseñanza-Aprendizaje Inteligentes. Tesis de Maestría. Universidad Central "Marta Abreu" de Las Villas.
- Méndez, G. y de Antonio, A. (2005). Training Agents: An Architecture for Reusability. IVA. LNAI 2650. Springer-Verlag Berlin Heidelberg 2005. p. 1–14.
- Minsky, M. (1986). The society of Mind. Simon & Shuster editors. New York.
- Novak, J. y Gowin, D. (1988). Aprendiendo a aprender. Ediciones Martínez Roca. Barcelona.
- Oliveira, F. y Viccari, R. (1996). Are learning systems distributed or social systems. in European Conference on AI in Education.
- Ontoria, A. (1993). Mapas conceptuales, una técnica para aprender. NARCEA, S. A. de Ediciones Madrid. 1993.
- Ovalle, D. y Jiménez, J. (2005). Sistemas de Enseñanza / Aprendizaje basados en Agentes Inteligentes Pedagógicos. Avances en Sistemas e Informática., Vol. 2(No. 2), p. 17-26.
- Peña, C. (2004). Intelligent agents to improve adaptivity in a web-based learning environment. Tesis de Doctorado. Department of electronics, computer science and automatic control. University of Girona.

- Ríos, L., Lezcano, M. y López, E. (2008). Un ambiente de aprendizaje asistido por computadora para la Programación Lógica. *Revista de Informática Educativa y Medios Audiovisuales*, 5.
- Rocha, R. (2005). IMPACT: Innovative Models for Prolog with Advanced Control and Tabling. *ICLP 2005, LNCS 3668, Springer-Verlag Berlin Heidelberg 2005*, 416 - 417.
- Rodríguez, M. (2007). Directorio Temático sobre la Programación Lógica. Tesis de Maestría. Centro Universitario de Sancti Spíritus "José Martí".
- Rodríguez, M., Estrada, V., J., F. y Castillo, J. (2008). El aprendizaje basado en problemas y el trabajo colaborativo como fundamentos para el diseño de cursos virtuales. Ejemplificación en la asignatura Informática II para las carreras de perfil empresarial. Congreso Internacional Universidad 2008, La Habana, Cuba. 2008.
- Russell, S. y Norvig, P. (1995). *Artificial Intelligence A Modern Approach*. Prentice-Hall Editor, A Simon & Schuster Company. ISBN: 0-13-103805-2. New Jersey, Estados Unidos.
- Schindlauer, R. (2005). Nonmonotonic Logic Programs for the Semantic Web. *ICLP 2005, LNCS 3668, Springer-Verlag Berlin Heidelberg 2005*, 446 - 447.
- Selker, T. (1994). Coach: A teaching agent that learns. *Communications ACM*, 37, 7, 92-99.
- Shalfield, R. (2005). Agent Toolkit. Documentación del software LPA WIN_ PROLOG.
- Silberschatz, A., Baer, P. y Gagne, G. (2005). *Operating System Concepts*. Seventh Edition. Bill Zobrist Editor. ISBN: 0-471-69466-5. Estados Unidos.
- Torrioni, P. (2003). Computational Logic in Multi-Agent Systems: recent advances and future directions. *Computational Logic in Multi-Agent Systems, Special Issue of Annals of Mathematics and Artificial Intelligence*, Kluwer Academic Publishers (2004).
- Vasconcelos, D., y Robertson, C. (2003). Rapid Prototyping of Large Multi-Agent Systems through Logic Programming. *Internacional Journal of AI in Education*. Springer-Verlag Berlin Heidelberg, 9, 51.
- Villarreal, G. (2006). Agentes Inteligentes en Educación. Centro Comenius Universidad de Santiago de Chile. EDUTEC. *Revista Electrónica de Tecnología Educativa*. No. 16.

Referencias Bibliográficas.

- Vizcarro, C. y León, J. (1998). Nuevas Tecnologías para el aprendizaje. Editorial PARANINFO.
- Weiss, G. (1999). Multiagent Systems. A Modern Approach to Distributed Modern Approach to Artificial Intelligence. Prentice-Hall Editor.
- Weiss, M. (2003). A gentle introduction to agents and their applications. MITEL Corp. Diponible en: <http://www.magma.ca/~mrw/agents/>.
- Wilson, W. G. (2005). Use of Logic Programming for Complex Business Rules. ICLP 2005, LNCS 3668, Springer-Verlag Berlin Heidelberg 2005, 14-20.

PRODUCCIÓN CIENTÍFICA DEL AUTOR SOBRE EL TEMA DE LA TESIS

Publicaciones en libros, revistas y memorias de eventos:

1. Ríos L., Lezcano M. y López E. Historia y evolución de los medios de enseñanza. Revista Iberoamericana de Educación. No. 37 de 2006. España
<http://www.rieoei.org/deloslectores/1166rios.pdf>
2. Ríos L., Lezcano M. y García A. Los mapas conceptuales, las TIC y el e-learning. Revista Iberoamericana de Educación. No. 37 de 2006. España.
<http://www.rieoei.org/expe/1570Rios.pdf>
3. Ríos L. y Lezcano M. Un ambiente de aprendizaje asistido por computadora para la Programación Lógica. Revista de Informática Educativa y Medios Audiovisuales (RIEMA). Argentina. Vol. 5, 2008
<http://laboratorios.fi.uba.ar/lie/Revista/Articulos/050510/A3mar2008.pdf>
4. Ríos L. y Lezcano M. Ambiente de autoaprendizaje asistido por computadora para la Programación Lógica. CD Memorias de Universidad 2008. 6to Congreso Internacional de la Educación Superior. La Habana, Cuba.
5. Ríos L., Lezcano M. y Díaz A. Mapas conceptuales y agentes inteligentes en la enseñanza del Prolog. CD Memorias del Segundo Congreso Cubano de desarrollo local. Universidad de Granma 2009. ISBN: 978-959-16-1004-1
6. Rodríguez M. y Ríos L. Plone un CMS que posibilita el desarrollo de aplicaciones Web para la educación virtual. VI Congreso Internacional de Informática en Salud. La Habana, febrero 2007.
http://www.informatica2007.sld.cu/Members/mirelyrh/plone-un-cms-que-posibilita-el-desarrollo-de-aplicaciones-web-para-la-educacion-virtual?set_language=en
7. Pérez Y. y Ríos L. Formación de Competencias profesionales en el Ingeniero Informático. CD-ROM II Evento Científico Metodológico sobre la enseñanza de las Ciencias Exactas. Holguín 2007. ISBN 978-959-18-0327-6.
8. Pérez Y. y Ríos L. Estrategia de enseñanza aprendizaje para la elaboración e implementación de un software educativo de modelación de Problemas de Programación Lineal. Boletín de la Sociedad Cubana de Matemática y Computación correspondiente al Volumen 3, No 1 del año 2005. ISSN 17286042

9. Ríos L. y Lezcano M. Rol del profesor ante las NTIC. CD del Segundo Evento Cubano COMPUMAT Espirituano 2003.
10. Ríos L. y Lezcano M. El rol de profesor en la sociedad de la información y la comunicación. Nuevas necesidades de formación del profesorado. Revista Electrónica INFOCIENCIA. ISSN 1029-5186. <http://www.magon.cu/mainpages/publishings.asp>
11. Ríos L., Lezcano M. y López E. Surgimiento y desarrollo de los medios de enseñanza. Revista Electrónica INFOCIENCIA Vol. 9 # 3, Septiembre, 2005. ISSN 1029-5186. <http://www.magon.cu/mainpages/publishings.asp>
12. Ríos L., Lezcano M. y García A. Convergencia de los Mapas Conceptuales, las TIC y el e-learning en el proceso de enseñanza-aprendizaje. Revista Electrónica INFOCIENCIA Vol. 10 # 3, Septiembre, 2006. ISSN 1029-5186. <http://www.magon.cu/mainpages/publishings.asp>
13. Pérez Y. y Ríos L. La Estadística Matemática a través de mapas conceptuales, sitio Web para apoyar el proceso de enseñanza-aprendizaje. CD ROM I Evento Internacional La matemática, la informática y la Física del siglo XXI. Holguín 2009 ISBN: 978-959-18-0498-3.
14. Ríos L. y Lezcano M. Un ambiente de enseñanza-aprendizaje inteligente para la Programación Lógica. CD-ROM Memorias del VII Congreso Provincial de Educación Superior Universidad 2010. Junio, 2009.
15. Ríos L., Lezcano M. y Díaz A. Enseñando Prolog usando mapas conceptuales. Revista Cubana de Ciencias Informáticas. Aceptado
16. Ríos L., Lezcano M. y Díaz A. Los mapas conceptuales y las TIC en la enseñanza de Prolog. Revista Avances en Sistemas e Informática. Colombia (esperando aceptación)

Otros eventos:

17. Pedagogía 2009, Provincial, Premio Relevante. Trabajo: Ambiente de enseñanza – aprendizaje asistido por computadora para la Programación Lógica.
18. EDUTEC 2008, Virtual. España. Trabajo: Un ambiente de enseñanza-aprendizaje para la Programación Lógica.
http://www.usc.es/tecnoeduc/edutec/otros_documentos/lista_acept.pdf

19. Forum Municipal de Ciencia y Técnica, 2008. Premio Relevante. Trabajo: Conjunto de Agentes Inteligentes para proporcionar adaptabilidad a APA-Prolog.
20. XV Forum de Ciencia y técnica, 2007. Premio Relevante. Trabajo: Ambiente de Autoprendizaje asistido por computadora para la Programación Lógica.
21. INFOCENTRO 2008. Premio al trabajo más relevante en Programa Audiovisual del MES. Trabajo: Agentes Inteligentes para proporcionar adaptabilidad a APA-Prolog.
22. INFOCENTRO 2007. Trabajo: Diseño e implementación de bases de datos para ambientes de enseñanza virtual.
23. COMPUMAT de las Provincias Centrales 2007. Premio Relevante. Trabajo: Los mapas conceptuales y las Tecnologías de la Información y las Comunicaciones.
24. COMPUMAT de las provincias Centrales 2004. Mención. Trabajo: Estrategias de enseñanza-aprendizaje para un software educativo de Programación.
25. Mujeres Creadoras, 2008. Premio Relevante. Trabajo: Ambiente de autoaprendizaje asistido por computadora para la Programación Lógica.
26. Encuentro Internacional Creatividad UPEL 2009. Venezuela. (Aceptado) Trabajo: Agentes inteligentes y mapas conceptuales en la enseñanza de la Programación Lógica.
27. Evento Provincial Universidad 2010. Premio Relevante. Trabajo: Ambiente inteligente de enseñanza-aprendizaje para Prolog.
28. Forum Municipal de Ciencia y Técnica, 2009. Premio Relevante. Conjunto de Trabajo: Inteligencia Artificial y Mapas Conceptuales para la enseñanza de Prolog.
29. COMPUMAT 2009. Evento Nacional. (Aceptado) Trabajo: Inteligencia Artificial y Mapas Conceptuales para la enseñanza de Prolog.

ANEXOS

Anexo 1. Encuesta para los Estudiantes

En varias ocasiones usted ha interactuado con el ambiente de enseñanza aprendizaje para la Programación Lógica (APA-Prolog). Le agradecemos de antemano que responda esta encuesta que pretende recoger sus opiniones acerca de la medida en que este le ayudó a comprender la asignatura. Le recordamos que las respuestas son anónimas por lo que le pedimos que sea sincero, ya que resultan muy importantes para perfeccionar nuestro trabajo.

1. ¿Consideras que la forma de representar el contenido que te ofrece APA-Prolog te ha facilitado comprender los conceptos claves de la Programación Lógica?
 definitivamente sí
 probablemente sí
 tengo dudas
 probablemente no
 definitivamente no
2. ¿APA-Prolog le ha ayudado a transformar las especificaciones de un problema dado en hechos, reglas y preguntas?
 mucho
 algo
 un poco
 muy poco
 nada
3. ¿Cómo evalúa la influencia del ambiente en su capacidad de expresar claramente un algoritmo empleando el estilo descriptivo?
 limitada
 discreta
 apreciable
 sobresaliente
4. ¿Qué aspectos le gustaría cambiar en el ambiente?
 - a) Ordene estos aspectos atendiendo a la prioridad que requieren para usted.
5. Evalúe la elevación de su nivel de conocimiento de la Programación Lógica a partir de la interacción con el ambiente.
 nivel alto
 medio alto
 medio
 medio bajo
 bajo
6. ¿Cree usted que ambientes similares a APA-Prolog facilitarían el proceso de enseñanza-aprendizaje de otros temas de otras asignaturas?
 sí no

Anexo 2. Respuesta de los estudiantes a las preguntas del cuestionario

Tabla A2.1 Respuestas a las preguntas 1, 2,3, 5 y 6

Pregunta 1	Pregunta 2	Pregunta 3	Pregunta 5	Pregunta 6
Probablemente sí	mucho	Muy poco	Medio	Sí
Definitivamente sí	algo	Apreciable	Medio alto	Sí
Definitivamente sí	mucho	Sobresaliente	Alto	Sí
Definitivamente sí	mucho	Discreta	Medio alto	Sí
Definitivamente sí	mucho	Sobresaliente	Alto	Sí
Definitivamente sí	algo	Apreciable	Medio alto	Sí
Definitivamente sí	algo	Apreciable	Medio	Sí
Definitivamente sí	mucho	Apreciable	Alto	Sí
Definitivamente sí	algo	Apreciable	Medio	Sí
Definitivamente sí	mucho	Sobresaliente	Alto	Sí
Definitivamente sí	algo	Apreciable	Medio alto	Sí
Definitivamente sí	mucho	Apreciable	Alto	Sí
Definitivamente sí	mucho	Apreciable	Medio alto	Sí
Definitivamente sí	mucho	Sobresaliente	Alto	Sí
Definitivamente sí	mucho	Sobresaliente	Alto	Sí
Definitivamente sí	mucho	Apreciable	Alto	Sí
Probablemente sí	mucho	Apreciable	Medio alto	Sí
Definitivamente sí	mucho	Apreciable	Medio alto	Sí
Definitivamente sí	algo	Apreciable	Medio alto	Sí
Definitivamente sí	mucho	Apreciable	Medio alto	Sí
Definitivamente sí	mucho	Apreciable	Medio alto	Sí
Definitivamente sí	mucho	Apreciable	Alto	Sí
Definitivamente sí	mucho	Apreciable	Alto	Sí
Definitivamente sí	mucho	Apreciable	Medio alto	Sí
Definitivamente sí	mucho	Apreciable	Medio alto	Sí

Tabla A2.2 Respuestas de los estudiantes a la pregunta 4

Encuesta 4	1. Me pierdo cuando entro a muchos conceptos y mapas.
Encuesta 6	1. Que hayan más ejemplos. 2. Una opción para los usuarios principiantes donde se les explique como navegar sobre el mapa.
Encuesta 7	1. Que sea más interactivo. 2. Que contenga más imágenes y diseños coloridos.
Encuesta 8.	1. Que contenga alguna imagen en movimiento. 2. Que tenga más colorido.
Encuesta 11	1. Debe tener un manual de usuario o guía para los usuarios principiantes.
Encuesta 18	1. Sería bueno que tuviera un espacio con los caminos a seguir para encontrar cada concepto.
Encuesta 19	1. Debe tener cosas que alegren el ambiente.
Encuesta 20	1. Que tenga un manual de usuario
Encuesta 25	1. Ponerle alegría al ambiente. 2. Un manual de usuario

Anexo 3. Tablas de frecuencia de las respuestas de los estudiantes a las preguntas de la encuesta.

Tabla A3.1 ¿Consideras que la forma de representar el contenido que te ofrece APA-Prolog te ha facilitado comprender los conceptos claves de la Programación Lógica?

		Frecuencia	Porcentaje
Válidos	Probablemente sí	2	8,0
	Definitivamente sí	23	92,0
	Total	25	100,0

Tabla A3.2 ¿APA-Prolog le ha ayudado a transformar las especificaciones de un problema dado en hechos, reglas y preguntas?

		Frecuencia	Porcentaje
Válidos	Algo	6	24,0
	Mucho	19	76,0
	Total	25	100,0

Tabla A3.3 ¿Cómo evalúa la influencia del ambiente en su capacidad de expresar claramente un algoritmo empleando el estilo descriptivo?

		Frecuencia	Porcentaje
Válidos	Limitada	1	4,0
	Discreta	1	4,0
	Apreciable	18	72,0
	Sobresaliente	5	20,0
	Total	25	100,0

Tabla A3.4 Evalúe la elevación de su nivel de conocimiento de la Programación Lógica a partir de la interacción con el ambiente.

		Frecuencia	Porcentaje
Válidos	Medio	3	12,0
	Medio alto	12	48,0
	Nivel alto	10	40,0
	Total	25	100,0

Tabla A3.5 ¿Cree usted que ambientes similares a APA-Prolog facilitarían el proceso de enseñanza -aprendizaje de otros temas de otras asignaturas?

		Frecuencia	Porcentaje
Válidos	Sí	25	100,0

Anexo 4. La moda y la mediana para las preguntas 1, 2, 3, 5 y 6 del cuestionario para los estudiantes.

Tabla A4.1 La moda y la mediana para las preguntas 1, 2, 3, 5 y 6 del cuestionario. (Corte 1)

	Average Linkage (Between Groups)					
	1		2		3	
	Mediana	Moda	Mediana	Moda	Mediana	Moda
Ha facilitado comprender los conceptos claves de la Programación Lógica	4,00	4,00	5,00	5,00	5,00	5,00
Ha ayudado a transformar las especificaciones de un problema	5,00	5,00	5,00	5,00	5,00	5,00
Influencia del ambiente en su capacidad de expresar claramente un algoritmo	2,00	2,00	4,00	4,00	4,50	4,00
Elevación de su nivel de conocimiento de la Programación Lógica	3,00	3,00	4,00	4,00	5,00	5,00
Facilitarían el proceso de enseñanza-aprendizaje de otras asignaturas	1,00	1,00	1,00	1,00	1,00	1,00

Tabla A4.2 La moda y la mediana para las preguntas 1, 2, 3, 5 y 6 del cuestionario. (Corte 2)

	Average Linkage (Between Groups)			
	1		2	
	Mediana	Moda	Mediana	Moda
Ha facilitado comprender los conceptos claves de la Programación Lógica	4,00	4,00	5,00	5,00
Ha ayudado a transformar las especificaciones de un problema	5,00	5,00	5,00	5,00
Influencia del ambiente en su capacidad de expresar claramente un algoritmo	2,00	2,00	4,00	4,00
Elevación de su nivel de conocimiento de la Programación Lógica	3,00	3,00	4,00	4,00
Facilitarían el proceso de enseñanza- aprendizaje de otras asignaturas	1,00	1,00	1,00	1,00

Anexo 5 Encuesta para la selección de los expertos que evalúan el Ambiente de Enseñanza-Aprendizaje para la Programación Lógica (APA-Prolog)

Método Delphi. Procedimiento de autovaloración de los expertos.

Agradeciéndole de antemano su colaboración en esta encuesta le solicitamos autoevalúe sus competencias como experto que dictaminará sobre el Ambiente de enseñanza-aprendizaje para la Programación Lógica (APA-Prolog).

Utilizaremos el procedimiento de autovaloración del método Delphi:

- a) Evalúe su competencia sobre el problema en una escala de 0 a 10. En esta escala el 0 representa que el experto no tiene competencia alguna sobre el problema y el 10 representa que posee una información completa sobre el tema.

0	1	2	3	4	5	6	7	8	9	10

- b) Marque, por favor, con una cruz cuál de las fuentes considera que ha influido en su conocimiento de acuerdo con el grado A, M o B.

ES DE ARGUMENTACION	FUENT	Grado de influencia de cada una de las fuentes en sus criterios.		
		A (alto)	M (medio)	B (bajo)
Conocimientos teóricos sobre la Programación Lógica.				
Su experiencia como profesor o profesional en el área de Programación de Computadoras				
Conocimiento sobre trabajos de autores nacionales relacionados con el tema				
Conocimiento sobre trabajos de autores extranjeros relacionados con el tema				
Conocimiento sobre metodología de enseñanza de la Programación				
Conocimiento sobre diseño de software educativo				

Los autores le quedan agradecidos por su colaboración y le garantizamos el anonimato.

Anexo 6 Determinación del nivel de competencia de los expertos y distribución por categorías docentes y científicas

EXP. #	KC	KA	K		F1	F2	F3	F4	F5	F6
1	0,8	0,95	0,875	ALTA	0,2	0,4	0,05	0,04	0,16	0,1
2	0,8	0,76	0,78	MEDIA	0,16	0,32	0,04	0,04	0,1	0,1
3	0,8	0,89	0,845	ALTA	0,2	0,4	0,05	0,04	0,1	0,1
4	0,7	0,88	0,79	MEDIA	0,2	0,4	0,04	0,04	0,1	0,1
5	0,8	0,82	0,81	ALTA	0,16	0,32	0,04	0,04	0,16	0,1
6	0,8	0,89	0,845	ALTA	0,16	0,4	0,05	0,04	0,16	0,08
7	0,8	0,92	0,86	ALTA	0,2	0,4	0,04	0,04	0,16	0,08
8	0,9	0,915	0,9075	ALTA	0,2	0,4	0,05	0,025	0,16	0,08
9	1	0,885	0,9425	ALTA	0,16	0,4	0,04	0,025	0,16	0,1
10	1	0,89	0,945	ALTA	0,16	0,4	0,025	0,025	0,2	0,08
11	1	0,99	0,995	ALTA	0,2	0,4	0,05	0,04	0,2	0,1
12	0,9	0,88	0,89	ALTA	0,16	0,4	0,04	0,04	0,16	0,08
13	0,6	0,635	0,6175	MEDIA	0,1	0,32	0,04	0,025	0,1	0,05
14	1	0,805	0,9025	ALTA	0,16	0,32	0,025	0,04	0,16	0,1
15	1	0,98	0,99	ALTA	0,2	0,4	0,04	0,04	0,2	0,1
16	0,7	0,98	0,84	ALTA	0,2	0,4	0,04	0,04	0,2	0,1
17	1	0,92	0,96	ALTA	0,16	0,4	0,04	0,04	0,2	0,08
18	1	0,74	0,87	ALTA	0,16	0,32	0,025	0,025	0,16	0,05
19	1	0,98	0,99	ALTA	0,2	0,4	0,05	0,05	0,2	0,08
20	1	0,98	0,99	ALTA	0,2	0,4	0,05	0,05	0,2	0,08
21	0,8	0,86	0,83	ALTA	0,2	0,4	0,04	0,04	0,1	0,08
22	0,6	0,785	0,6925	MEDIA	0,2	0,32	0,04	0,025	0,1	0,1
23	0,8	1	0,9	ALTA	0,2	0,4	0,05	0,05	0,2	0,1
24	0,8	0,9	0,85	ALTA	0,16	0,4	0,04	0,04	0,16	0,1
25	0,8	0,885	0,8425	ALTA	0,16	0,4	0,04	0,025	0,16	0,1
26	0,8	0,79	0,795	MEDIA	0,16	0,32	0,025	0,025	0,16	0,1
27	1	0,99	0,995	ALTA	0,2	0,4	0,05	0,04	0,2	0,1
28	0,6	0,53	0,565	MEDIA	0,1	0,2	0,025	0,025	0,1	0,08
29	0,9	0,965	0,9325	ALTA	0,2	0,4	0,04	0,025	0,2	0,1
30	0,9	0,97	0,935	ALTA	0,2	0,4	0,04	0,05	0,2	0,08

F1-F6: Fuentes de Argumentación.

KC: es el coeficiente de conocimiento o información que tiene el experto acerca del problema calculado sobre la base de la valoración del propio experto

KA: es el coeficiente de argumentación o fundamentación de los criterios del experto determinado como resultado de la suma de los puntos alcanzados a partir de la tabla patrón (Anexo 8):

K: Coeficiente medio.

Categoría Docente	Cantidad	%
Titulares	16	53.3
Auxiliares	14	46.7
Categoría Científica		
Doctores	22	73.3
Master	8	26.7

Anexo 7. Encuesta para los Expertos

En el Centro Universitario de Sancti Spíritus “José Martí” se ha desarrollado un ambiente de enseñanza- aprendizaje para la Programación Lógica (APA-Prolog). En este momento pretendemos recoger opiniones acerca de la calidad de este recurso. Hemos pensado en usted por el dominio que tiene sobre el tema, le sugerimos que al revisar el software fije su atención en el contenido, la armonía, el tratamiento diferenciado a los estudiantes y en la conveniencia de extender la experiencia a otras asignaturas. Le agradecemos de antemano su colaboración, le garantizamos el anonimato y le pedimos que sea sincero, considerando la importancia de sus criterios para perfeccionar este trabajo.

1. ¿Considera que a partir de este ambiente es posible elevar los niveles de asimilación de la Programación Lógica?
 muy adecuado
 adecuado
 ni adecuado ni inadecuado
 inadecuado
 muy inadecuado
2. ¿Aprecia usted, entre todos los recursos que conviven en el ambiente, una armonía que propicie que los estudiantes se apropien de los contenidos de la asignatura?
 nada
 poco
 mucho
3. Su valoración acerca de las posibilidades que tiene el ambiente de tomar en cuenta los conocimientos previos del estudiante es:
 muy buena
 buena
 ni buena ni mala
 mala
 muy mala
4. De los recursos disponibles en APA-Prolog ¿cuáles reconoce de mayor utilidad para conseguir que los alumnos eleven sus niveles de asimilación? (Marque con una X)
 directorio temático () simuladores ()
 espacio para el debate () entrenadores ()
 conversación en línea () evaluadores ()
 textos explicativos ()
a) Ordene los recursos seleccionados atendiendo a la calidad y potencialidad dentro del ambiente desarrollado.
5. ¿Cree usted que ambientes similares a APA-Prolog facilitarían el proceso de enseñanza-aprendizaje de otros temas de otras asignaturas?
 sí no no se

Anexo 8. Importancia asignada por los expertos a los recursos

Directorio temático

		Frecuencia	Porcentaje
Válidos	1,00	12	40,0
	3,00	2	6,7
	4,00	3	10,0
	5,00	10	33,3
	6,00	2	6,7
	7,00	1	3,3
	Total	30	100,0

Espacio para el debate

		Frecuencia	Porcentaje
Válidos	1,00	11	36,7
	3,00	1	3,3
	5,00	6	20,0
	6,00	11	36,7
	7,00	1	3,3
	Total	30	100,0

Conversación en línea

		Frecuencia	Porcentaje
Válidos	1,00	11	36,7
	4,00	1	3,3
	6,00	5	16,7
	7,00	13	43,3
	Total	30	100,0

Textos explicativos

		Frecuencia	Porcentaje
Válidos	1,00	20	66,7
	2,00	7	23,3
	3,00	1	3,3
	4,00	2	6,7
	Total	30	100,0

Simuladores

		Frecuencia	Porcentaje
Válidos	1,00	18	60,0
	2,00	8	26,7
	3,00	1	3,3
	4,00	2	6,7
	5,00	1	3,3
	Total	30	100,0

Entrenadores

		Frecuencia	Porcentaje
Válidos	1,00	12	40,0
	2,00	4	13,3
	3,00	11	36,7
	4,00	2	6,7
	6,00	1	3,3
	Total	30	100,0

Evaluadores

		Frecuencia	Porcentaje
Válidos	1,00	12	40,0
	3,00	3	10,0
	4,00	9	30,0
	5,00	2	6,7
	7,00	3	10,0
	Total	29	96,7
Perdidos	Sistema	1	3,3
Total		30	100,0